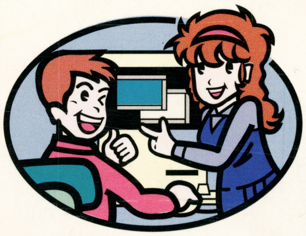
2001年9月1日発行（毎月1回1日発行）第20巻第9号 昭和57年12月6日第三種郵便物認可

マイコン

わかる！動かせる！
プログラムが組める雑誌

BASIC Magazine

SEP.2001 9



定価 550Yen



『BASIC STUDIO』プロデューサーが教える

実践編 PS2でプログラム！ シューティング・ゲームの制作

編集部の学校訪問記

埼玉県立春日部東高等学校





エレクトロニック研究所訪問

東京大学生産技術研究所
東京大学人工工学研究センター

好評連載

プログラマー養成講座

Technical News Spot
速度が上がるとパソコンの使いかたが劇的に変わる！

はじめようMicrosoft ExcelでのBASIC
toto予想プログラムにチャレンジ（1）

DirectXテクノロジーを伝授する！
スプライト・アニメーションでものを動かす！

図解！Visual Basicゲーム作成術
敵キャラクターに弾を撃たせる！

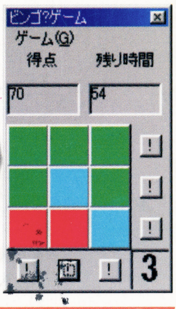
PCでロボットを動かす
BN-1わがままカプリロ＋プログラム・キット
レゴ・マインドストームでライン・トレーサー

マンガで学ぼう/パソコンレクチャー
驚異の指コンピュータ

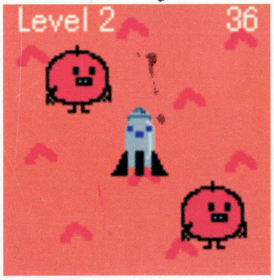
読者が作った作品をドーンと紹介

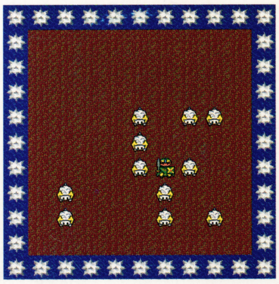
特選投稿プログラム・コーナー













VisualBasic6.0	Word Hunter
VisualBasic6.0	Sky Fight
VisualBasic6.0	ビンゴ?ゲーム
VisualBasic6.0	(超大作) FUKUWARAI 2
VBScript+DirectX	(超大作) ピクチャVer.1.04
C++Builder4	らむしふと
HSP V2.5	シャボン狩り
Javaアプレット	ヌー星人
Qbasic	(移植版) Mouse対応版 SHORTEN
Qbasic	Paper Plane Forever
WonderWitch	スネーク

CMM試聴室

SC-8820	スプリング・ラブ
SC-88Pro	Moon Rider

専用ホームページ
http://www.basicmagazine.net/

マイコン BASIC Magazine

CONTENTS

9

SEP.2001

専用ホームページ　http://www.basicmagazine.net/

広告さくいん

©マイコンBASICマガジン9月号
第20巻 第9号 2001年8月25日印刷
2001年9月1日発行
定価550円(本体524円)
電波新聞社　2001 Printed in Japan
編集発行人：平山哲雄
東京都品川区東五反田1-11-15
郵便番号141-8715
電話　東京(03)3445-6111（大代表）
振替　（東京）00150-3-51961
印刷：奥村印刷(株) 製本：(株)堅省堂

ITを主軸としたエレクトロニクス
最先端技術と製品に出会える展示会

シーテック ジャパン
CEATEC JAPAN 2001

ジュニアサイエンススクール
―わくわくロボット研究室―開催！

幕張メッセで開催される「CEATEC JAPAN 2001」にて，10月6日(土)，小学校高学年(4～6年生)を対象とした「ジュニアサイエンススクール」が行われます。

これは，次世代のテクノロジーや科学の担い手である子供たちに，楽しみながら最新の技術にふれてもらおうというイベントです。

宇宙や生産現場，家庭などで人間を手助けし，あるいは遊び相手にもなってくれるロボット。現実的にもますます身近なものになっています。会場には，ロボット研究開発の最前線の大学研究室から，さまざまなロボットが大集合。研究者自身が，ロボットの歴史やその役割，人との関係などを熱く語ってくれます。

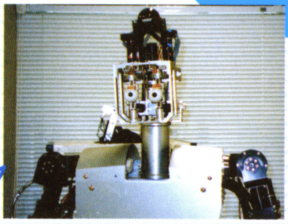
同時にロボットの分解や組み立ても実演。外部からの情報に反応して動くロボットの仕組みが，人間の体の仕組みとも密接に関係していることなどを楽しく学ぶことができます。

家族や学校単位で参加してみてはいかがでしょう。

参加者を募集中！

対象：小学校高学年（4～6年生）

会場で会える電気通信大学のロボットたち



▲コミュニケーションロボット



▲ホーミング犬ロボット

▼チアリーディングロボット



●募集要項

- 開催期日：10月6日（土）
  1回目13:30～14:30　　2回目15:30～16:30
- 開催場所：幕張メッセ・CEATEC JAPAN会場内
- 参加対象：小学4年生～6年生
- 参加費：無料
- 参加方法：下の参加申込書に記入しFAXで申し込んでください。FAX番号03-3423-1601
- 申し込み締切：9月7日（金）
- 招待券プレゼント：申込者全員に「CEATEC JAPAN 2001」の入場券を4枚プレゼント
- 問い合わせ先：ジュニアサイエンススクール事務局（株）インターグループ　担当＝小田・中河原
  TEL03-3479-5131

学校の先生方へ

最先端の情報，テクノロジーにふれることのできる絶好の機会です。課外授業や社会見学などの一環としてご利用ください。

※CEATEC JAPAN 2001開催情報は右ページをご覧ください。

「電子工作教室」連日開催

電子工作キットをハンダコテやドライバーなどを使って組み立て体験ができるというもので，昨年も大好評でした。CEATEC JAPAN 2001会期中行われており，当日自由参加となります。



▶親子連れなどでにぎわった昨年の「電子工作教室」のもよう

ジュニアサイエンススクール　参加申込書

ジュニアサイエンススクール事務局までFAXでお申し込みください。

FAX：(03)3423-1601　締切：9月7日(金)

フリガナ			
氏　名		性別	男　・　女
フリガナ			
学校名	区・市・町・村・私　立	小学校	年　生
自宅住所	（〒　　　―　　　）		
電　話	（　　　　）　　　―		
希望時間に○印を付けてください。	午後1時30分開始の部	・	午後3時30分開始の部

エレクトロニック研究所訪問

東京大学生産技術研究所
東京大学人工物工学研究センター

東京大学　http://www.u-tokyo.ac.jp/



天下の東大！

「東大（とうだい）」でおなじみの東京大学ですが，その東京大学駒場リサーチキャンパス（東京都目黒区駒場）には，生産技術研究所，先端科学技術研究センター，気候システム研究センター，国際・産学共同研究センター，人工物工学研究センター，空間情報科学研究センター，先端経済工学研究センターなど，数多くの研究所・研究センターが存在します。

今回はその中から，人工物工学研究センターと生産技術研究所を取材してきました。

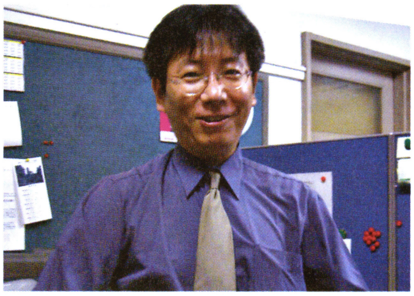
東京大学人工物工学研究センター（RACE）

http://www.race.u-tokyo.ac.jp/

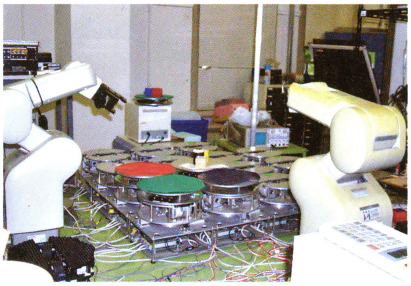
人工物工学研究センターの発足は1992年。この研究センターは教官・研究員が総勢で10数名（院生を含むと30名程度）と比較的小人数ですが，「人工物の持つ意味と，今後どのように存在すべきかを領域横断的に考察する」ことをテーマに知能のシステム化，インテリジェントCAD，マテリアルデザイン，3次元情報のデータ圧縮，ロボット制御などを研究しています。



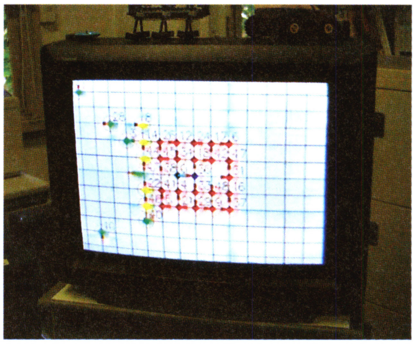
人工物工学研究センターは，ツタの茂った16号館の中に



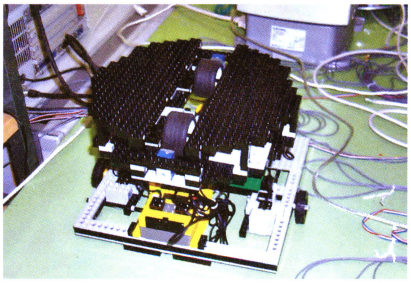
下村研究室の下村芳樹助教授。「自律分散制御によるパターン形成技術」や「自己修復システムの開発」などを研究されている



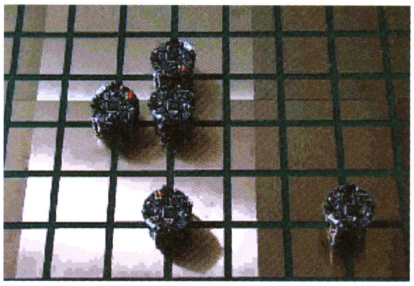
冨山研究室（冨山哲男教授）と共同研究テーマ。多数の自律型ロボットによって構成される「やわらかい機械：細胞型生産システム」



マイクロマシンの動きをシミュレーションしたようす。丸の一つ一つがマイクロマシンに相当



LEGO MINDSTORMSで製作中のロボット。入出力ポートが足りない問題をどうすればよいか検討中とのこと



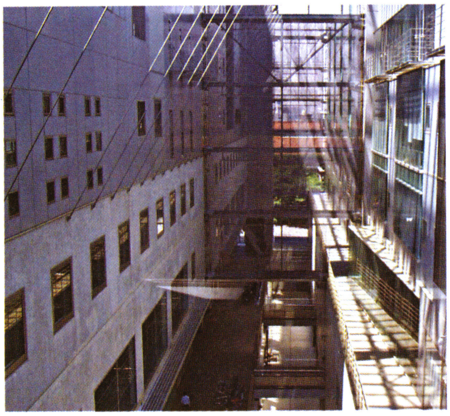
「マイクロマシン群のための作業パターン形成技術」という研究テーマで川崎重工業と共同開発された自律分散制御システム。故障した機体を排除し，素早く形態を修復する自己修復機能を搭載



CGアートで有名な河口洋一郎教授（情報造形学）。オリジナルの造形・運動アルゴリズムから生み出される映像は唯一無二

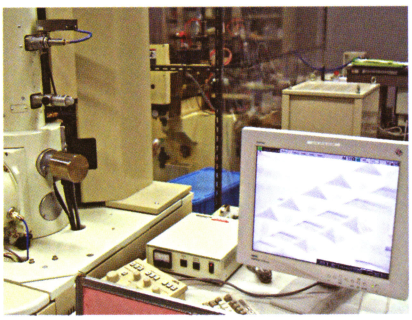
東京大学生産技術研究所（IIS）

http://www.iis.u-tokyo.ac.jp/

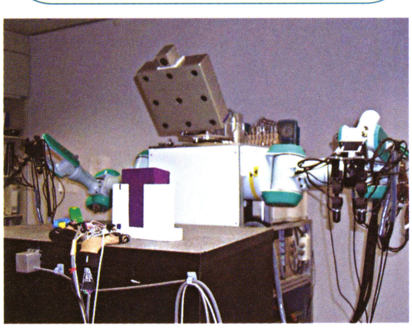


東京大学生産技術研究所は，1949年に東京大学の附置研究所として発足しました。工学領域のほぼ全分野をカバーし，約340名の教職員が在籍しています。

研究棟は地上8階，地下1階。その最新の巨大建築は圧巻です。



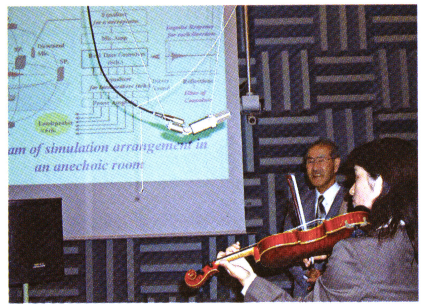
μｍ単位で加工した部品を電子顕微鏡で表示（マイクロメカトロニクス国際研究センター）



人の動きのマネするという人間行動観察ロボット（情報・システム部門，情報統合研究群，池内研究室）



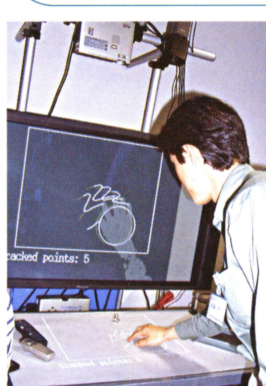
200台のPCを結合した大規模クラスタで超並列WEBウエアハウスサーバを構築（概念情報工学研究センター，喜連川研究室）



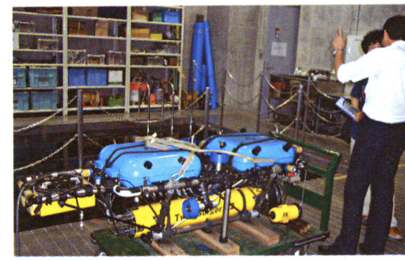
橘研究室（人間・社会部門，都市空間研究群，応用音響工学）の遮音性能をもつ材料で囲まれた無響室



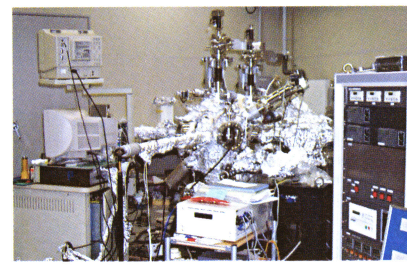
佐藤研究室（概念情報工学研究センター）では「視線方向の利用」という新しいヒューマン・コンピュータ・インタラクションを研究中



佐藤研究室の佐藤洋一助教授に拡張机型インターフェース「EnhancedDesk」を実演していただく。指先の認識は上方の赤外線カメラで行っている



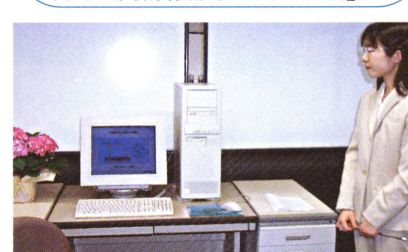
浦研究室（海中工学研究センター，海中ロボット学）の「自律海中ロボット TWIN-BURGER 2」



平川研究室（物質・生命部門，次世代デバイス研究群）のテーマは「量子半導体エレクトロニクス」。写真は超高移動度ヘテロ構造，量子ドット，超格子構造などの結晶を成長させるのに使われる「分子線エピタキシー装置」



池内研究室（情報・システム部門，情報統合研究群，視覚情報工学）による「実空間データ収集車による3次元空間情報都市地図の生成」



吉川研究室(情報・システム部門，構造システム研究群）の「凸包モデルを用いた最悪地震震動の推定」

最後に

一つ一つを丁寧に見ていくと時間が足りなくなってしまうほど，多くの研究テーマがありました。さらに今回は七つの研究所・研究センターのうち2か所しか回れませんでしたので，その規模の大きさには驚くばかりです。こうした設備はオープンキャンパスで年に1回，一般公開されていますので，そうした日を利用して実際に足を運んでみてはいかがでしょうか。



▲3年生・理系クラスのみなさん



▲1年生のみなさん

編集部の学校訪問記

埼玉県立 春日部東高等学校

http://www.higashi-hs.kasukabe.saitama.jp



▲物理部のみなさん



春日部東高校は，昭和52年4月に開校された比較的新しい学校です。普通科のほか，平成6年度に関東で初めて設置された人文科の二つの学科があり，人文科開設を機に教育内容のより一層の充実を目指しています。

勉学だけではなく部活動の活躍も目覚ましく，陸上部の全国大会出場や物理部の埼玉県科学教育振興展覧会中央展での優秀賞など，実績を積み重ねています。



▶情報教育にかかせない鈴木成先生

「GroupWise」導入で情報活用能力が向上

同校では，生徒の情報活用能力を高めるための教育が積極的に行われています。その一環として，平成4年からネットワークを構築しはじめ，平成9年に全館の約80台のパソコンがLANで接続されました（全長1kmにもおよぶ配線）。平成10年には校内のパソコンを一新，ノベルのグループウエア「GroupWise」を導入しました。

「サーバ側で生徒のアカウント管理が簡単にできるので，安心して生徒の電子メール活用を促進できます」と話すのは，パソコン関連の導入からシステム管理までを担当している鈴木　成先生。準備室に設置されているサーバは，メールサーバ／ファイアーウォール／ファイルサーバ／成績データベース／FAX用／インターネット（2台）の7台。鈴木先生をはじめ，情報教育を担うシステム管理担当5名の先生が助け合いながら生徒をサポートしています。



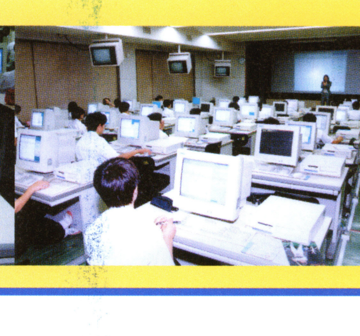
▲準備室にある各種のサーバ

現在はOCNエコノミーによる常時接続（128k：32,000円／月）を使用したインターネットですが，文部科学省と総務省の共同プロジェクトによる「学校インターネット」の研究指定校になっており，今年10月には10Mの専用回線が引かれるそうです。

パソコンを使った授業は視聴覚室で

視聴覚室には，Pentium200 MHz搭載のIBM GL300（OSはWindows95）が42台（生徒用）。先生側のパソコンの画面が大型プロジェクタで映し出されます。





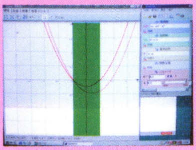
1年生の「数学の授業」を見学

1年生の数学の授業では「前回の授業で解いた問題をコンピュータで確かめてみよう」というテーマで，“2次関数のグラフがともにx軸と共有点をもつとき，定数mの値の範囲を求めよ”という解答をグラフを使って検証していました。

使用しているソフトは，数学担当の渋井雅枝先生が自ら探してきたという，フリーソフトの「GRAPES」。

渋井先生はパソコンを使う良さに関して「紙で答えた解答は正確だが，イメージがわかない。コンピュータでイメージを植え付ける」と話しています。


▶渋井雅枝先生


◀これが，「GRAPES」で描かれたグラフ


◀熱心にパソコンで検証する生徒たち

3年理数系の「政経の授業」を拝見

3年・理数系の政経の授業は，「一部娯楽的な部分がある」（菅原和浩先生）ということですが，「公害問題を知ろう」というテーマで，インターネットを駆使して　・足尾銅山鉱毒事件とは　・田中正造代議士はどんな活躍をしたか　・典型7公害とは　・環境基本法の成立とその意義…などを調べ，後でレポートにまとめて提出するのだそうです。


◀検索のしかたなどを指示する菅原先生


▶ちゃんと勉強してるかな？

MSX2を使った授業も！

2年生理科の授業では，理科室にあるMSX2でモーショントラックのプログラム（鈴木先生が組んだ）を使い，ゲーム感覚で「力と運動の変化」を学んでいました。

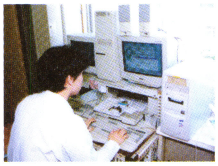
鈴木先生いわく「MSX2といえども，まだまだ使えるパソコン。使い道が限定されるが，使えるものはうまく利用している」ということです。

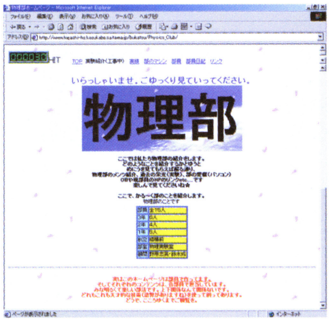


数々の実績を残している「物理部」

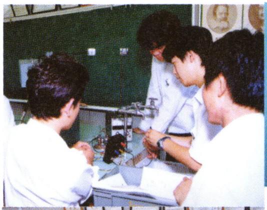
これまで，いろいろな展示会やコンテストに作品を応募し，実績をあげている物理部（鈴木先生顧問）が活動している物理実験室を拝見しました。埼玉県科学教育振興展覧会や，埼玉県理科教育研究発表会などに作品を出展し，優秀賞などを受賞している活発な部です。現在15人の部員がおり，今年も，「弾性体を伝わる三角波」をテーマにした実験作品を「全国高等学校理科・科学クラブ研究論文」に出展しました。

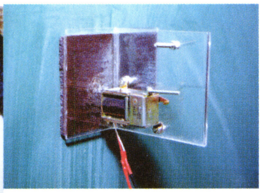
実験はパソコンと連動し，実験装置を制御したり，データをExcelなどでまとめ，解析したりしています。

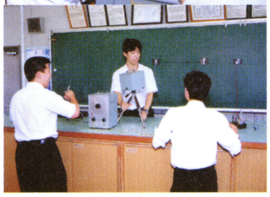

▲C言語で組んだプログラムで制御



▲生徒が作り，更新している物理部のページ






◀▲取材時に見せてくれた「全国高等学校理科・科学クラブ研究論文」に出展中の「弾性体を伝わる三角波」実験。フロッピードライブの磁石を利用している


▲数々の実績を記した賞状

ホームページ



春日部東高等学校　〒344-0002 埼玉県春日部市樋籠363　☎048-761-0011

中学生・高校生を対象にした「ホームページコンテスト」応募作品募集中

財団法人　国際コミュニケーション基金（ICF）主催の『第4回ICFホームページコンテスト』にあたり，現在作品を募集中です。これは，中学生・高校生を対象にしたもので「自分発，新コミュニケーション」をテーマにしたコンテストです。個人，団体問わず応募可能です。

同コンテストには電波新聞社も後援しています。どしどし参加してみましょう。

- ●目的：自己表現を通じてインターネット時代のコミュニケーションを考える
- ●応募期間：2001年9月30日（日）まで
- ●応募資格：中学生・高校生
- ●応募方法：ホームページ上でURLを登録
  http://www.icf.or.jp/contest/
- ●審査：独自性，コミュニケーションへの意思，双方向性，国際性，質・量，見やすさ・扱いやすさなどを基準に，1次審査・2次審査により総合的に評価。
- ●審査発表：2001年10月19日（金）ホームページ上で発表
- ●表彰：個人の部，団体の部それぞれ最優秀賞（1）優秀賞（2）特別賞（1）入賞者にはコンピュータ周辺機器を贈呈
- ●表彰式：2001年11月10日（第2土曜日）午後／東京都新宿区のKDDIホテルストラーダ
- ●問い合わせ先：
  e-mail　contest@icf.or.jp
  fax 03-5202-8509

わかる！おもしろい！初歩からのプログラミング講座

プレステ2でゲームをプログラムする！



実践編：シューティング・ゲームの制作

「BASIC STUDIO」アシスタントプロデューサー ●アートディンク 齋藤博希

前号のあらすじ

前号では，BASICに命令を与えたところからプログラムの作成，そして画面上を左から右に文字を移動させるプログラムまでを行いました。

今回は，前回説明した内容に＋αして簡単なシューティング･ゲームを制作してみたいと思います。

何を作ろう

ゲームを楽しんだことのある人なら一度は，自分でゲームを作りたいと思ったことがあるのではないでしょうか？そのとき，あなたはどんなゲームを作りたいと思いましたか？ちょっと思い出してみてください。

どのようなゲームが作りたい？と聞いてよく返ってくる回答が「○○みたいなゲームが作りたい」というもの。しかし，この答えには重大な問題点があるのです。

くわしく考えよう

「○○みたいなゲームを作りたい」の○○に有名なゲームを当てはめると，答えを聞いた人にも作りたいゲームの形がおぼろげに見えてくるかもしれません。しかし，○○に当てはまる言葉がマイナーなゲームだったりしたら，どんなゲームが作りたいのかまったく見当もつきませんね。

この回答は聞く人が○○に当てはめるゲームについて知っているということが条件になってしまいます。ゲームを知らない人にもわかるように説明するには，この回答ではいけないことになります。

最近のゲームの制作状況では，ご存知のようにすべての行程を1人で作成することはなく，プログラムを組む人でも複数いるような分業体制となっています。このようなとき，担当者に作業を行ってもらうのに，ゲームの内容を細かく説明できる必要があるのです。

え，「どうせ自分1人で作るのだから自分がわかっていればいいよ」ですって？確かに作るのはあなたですが，実際にプログラムを処理してくれるのは他ならぬBASICです。残念ながらBASICはゲームに関する知識などまったくないので，プログラムを書く私たちが細かく説明しなければならないのです。

要素を分解しよう

細かく説明，といっても何から手をつけてよいのかわからないと思います。まずは手始めに簡単なシューティング・ゲーム（以下STG）を例にとって説明してみましょう。

まず，作ろうとするSTGの内容を大まかに書き表してみます。

- コントローラの方向キーで自機が移動する。
- コントローラの○ボタンで自機から弾が発射される。
- 画面上部から敵が画面下部に向かって飛んでくる。
- 敵に自機が当たるとミス
- ミスが３回でゲーム・オーバー
- 自機の弾が敵に当たると敵は消滅し，得点が入る。

とりあえず，このような感じでしょうか？これで大まかなゲームの流れが思い浮かぶと思います。要はあなたの頭の中でプレイしたゲームの内容を一つ一つ書きだせばよいのです。

なお，はじめはこれぐらい大まかで構いません。あまり細かいところを書き出していると，きりがありませんし，作っているうちに，新しいことを思い付いたり，予想外の事体が発生することもあります。

要素をさらに分解する

さて，今度は各要素をさらに細かく分解します。たとえば，先ほどの「方向キーで自機が移動」のところを，このように分解します。今回はテキスト文字を使用するので，自機を文字の「A」で表示することにします。すると

- 方向キーの右が押されたら自機を右に1文字分移動
- 方向キーの左が押されたら自機を左に1文字文移動
- 方向キーの上が押されたら自機を上に1文字分移動
- 方向キーの下が押されたら自機を下に1文字分移動

といった形に分解できます。

なんだか同じようなものが並んでいますが，これが実際の手順として使用することになります。

関数について

さて，先ほど分解した要素で処理を実行するには，方向キーが押されたかどうかを判断する必要があります。方向キーが押されたかどうかなどのコントローラの状態を知るには，関数というものを使います。関数とは数字や文字列をわたすことで，別の数値や文字列に変更してくれる働きをするものです。まずは基本的な関数を使って関数について説明することにします。

```
PRINT IP(1.25)
```

上の文を実行すると，「1」が表示されます。PRINTの後に書かれた「IP」が関数と呼ばれるもので，その後の「()括弧」の中に書かれた数値が引数と呼ばれます。IP関数は引数に書かれた数値の整数部分のみを返す関数です。つまり，「1.25」という数値がIP関数によって小数点以下が切り捨てられ，1という整数に変更されたのです。

関数の基本的な書きかたは，つぎのようになります。

```
関数(引数)
```

関数によっては，引数がない物や引数をいくつかわたさなければならないものがあります。また，引数が数値か文字列かは関数によってちがうので注意しましょう。

コントローラの状態を取得する

それではここでコントローラの状態を取得する「PAD」関数を使ってみましょう。

PAD関数はコントローラ番号とボタン番号の二つの引数によって指定されたボタンの押された強さを数値として返してくる関数です。まずはPAD関数がどのようなものか見てみましょう。

```
10 DO
20 PRINT PAD(1,1)
30 LOOP
```

コントローラ端子1に接続されたコントローラの方向キーの右を押すと画面上に表示されている数値が0～255の間で変化します。方向キーの右ボタンを押していなければ「0」が表示され，強く押すほど数値が大きくなるのがわかると思います。引数とボタンの対応については**第1図**を見てください。

文字をコントローラで動かそう

さて，PAD関数の使いかたがわかったところで，PAD関数を使って自機を動かすプログラムを作ってみましょう。先ほど，文字を動かすことについて，要素を分解しました。ここではこの要素をBASICでわかるように翻訳してみたいと思います。

例として

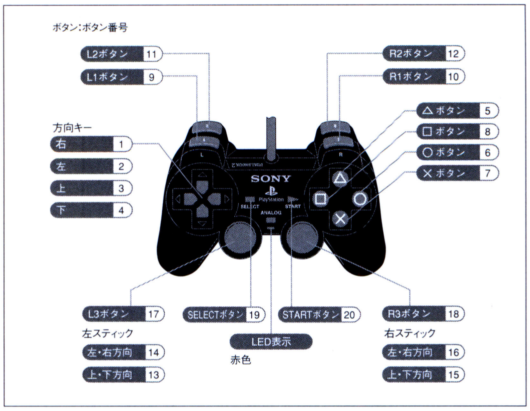
- **方向キーの右が押されたら自機を右に1文字分移動**

をBASICで翻訳してみることにします。

まず，方向キーの右が押されたということは，PAD（1,1）を使って，0より大きい値が返ってきたときと判断することができます。

また，自機であるAを右に1文字分移動するには，前回の講義の最後に行ったように，文字の横座標を示す変数に+1して新しい位置に同じ文字を表示することで表現することができます。よって，先ほどの要素はBASICでこのように言い換えることができます。

PAD（1,1）が0より大きければ，
自機の横座表の値を+1する。
自機の座標に文字「A」を表示



《第1図》 引数とボタンの対応

それでは，早速この要素をBASICでプログラムとして書いてみることにします。

自機である文字「A」を表示するための座標を(PX,PY)で表すとして，

```
10 DO
20  CLEAR
30  IF PAD(1,1)>0 THEN PX=PX+1
40  LOCATE PX,PY
50  PRINT "A"
60  FSYNC 1
70 LOOP
```

となります。

よくわからない場合は，先月号の最後にある文字の移動プログラムと見比べてみましょう。どこがちがうかを比べることで，よりプログラムを理解することができます。

さて，先ほどのプログラムで，方向キーの右を押し続けていると，「座標指定の誤りです」「指定位置が画面範囲を超えています」といったエラーが表示され，プログラムの実行が止まってしまいます。これは，LOCATE命令で画面の外の座標を指定すると，「そこにカーソルを移動することができないよ」と注意されてしまうのです。このような現象を回避するにはどうしたらよいでしょうか？

《リスト1》 自機の移動を表現する文字移動のプログラム・リスト

10 DO	繰り返しはじめ
20　CLEAR	画面上の文字を消去
30　IF PAD(1,1)>0 THEN PX=PX＋1	方向キーの右が押されていれば、座標用変数Xに1加える
40　IF PAD(1,2)>0 THEN PX=PX－1	方向キーの左が押されていれば、座標用変数Xから1ひく
50　IF PAD(1,3)>0 THEN PY=PY－1	方向キーの上が押されていれば、座標用変数Yから1ひく
60　IF PAD(1,4)>0 THEN PY=PY－1	方向キーの下が押されていれば、座標用変数Yに1加える
70　IF PX<0 THEN PX=0	X座標が画面より左を指定しないようにする
80　IF PX>36 THEN PX=36	X座標が画面より右を指定しないようにする
90　IF PY<0 THEN PY=0	Y座標が画面より上を指定しないようにする
100　IF PY>22 THEN PY=22	Y座標が画面より下を指定しないようにする
110　LOCATE PX, PY	テキスト座標のX，Yにカーソルを移す
120 PRINT ″A″	「A」を表示する
130 FSYNC 1	画面書き換えを1回待つ
140 LOOP	DOに戻る

エラーを回避するにはこのようにします。

① **30行でXの値を増やした後にXの値がLOCATEで指定できるX座標の最大値を超えていないか判断する。**

② **最大値を超えていたら，Xの値を最大値に直す。**

手順の①はIF文を使うことでBASICに翻訳できそうです。また，②はXに最大値を代入することで行います。

では，先ほどのプログラムにこの1行を付け加えてみましょう。

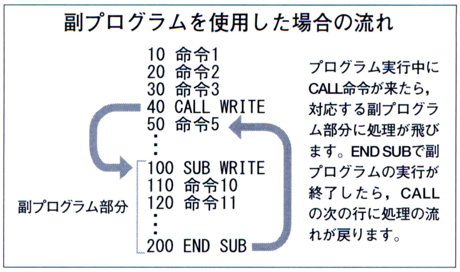
```
35  IF PX>=37 THEN PX=37
```

これで右を押し続けてもエラーで止まることはありません。あとは左，上，下の処理を同じように加えるだけで，自機の移動を表現する文字移動のプログラムは完成です(**リスト1**)。

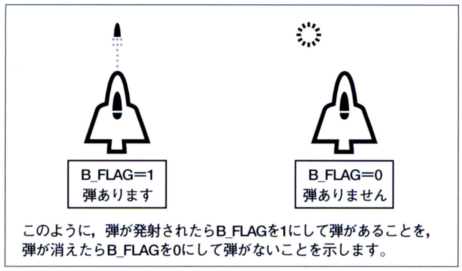
なお，先ほどとちがい，横座表を表すXの最大値がテキスト座標の最大値である37ではなく，36であるのは理由があります。私たちがキーボードから文字を入力したときと同じように，PRINT文で文字を表示すると，カーソルの位置は一つ分右にずれます。このため，画面右端で文字を表示すると，場所によっては文字が思った場所に表示されなくなってしまうのです。興味のある人は，80行のX座標の制限を37にして実行してみてください。

副プログラム化する

先ほどのプログラムではメイン・ループとなるDO～LOOPの中に「コントローラ入力による自機の移動」と「自機の表示」を入れて実行しています。このまま，メインループにその他の処理を書き加えてもいいのですが，そうした場合，メインループがごちゃごちゃしてしまい全体の流れが見にくくなります。そこで，「副プログラム（サブ・ルーチン）」を使ってメインループを簡略化します。



《第２図》 副プログラムの概念



《第３図》 フラグの考えかた

副プログラムとは，よく使う処理をまとめて名前をつけておき，そのプログラムを使いたいときに呼び出せばどこからでも，何度でも呼び出すことのできる物です（第２図）。

副プログラムを作成する

副プログラムを作成するには,「SUB」命令と「END SUB」命令で副プログラム化したい部分を挟んで作成します。このとき，SUB命令の後にパラメータとして副プログラム名を付けます。

SUB 副プログラム名
∫
副プログラム部分
∫
END SUB

作成した副プログラムはCALL文で呼び出して使うことができます。逆に呼び出さないと実行されません。基本的な副プログラムの使用例を，先ほどの自機移動のプログラム（**リスト１**）を副プログラムを使って書き直した物を**リスト2**に示しますので参考にしてください。

弾の処理

それでは，つぎに自機から弾を発射するプログラムを作ってみましょう。まずは自機の移動と同様に，弾を発射するという要素を細かく分解してみることにします。

①○ボタンが押されたら，自機の前方から弾を発射する
②発射された弾はまっすぐ上に進む
③画面の外に移動するか，敵に当たると弾は消える

大体こんな感じでしょうか？自機の移動プログラムとの大きなちがいは，つぎの2点が挙げられます。

A.○ボタンを押してから，弾が画面の外に移動するか，敵に当たるまでの間しか，弾は表示されない。
B.弾が発射されたら，コントローラの操作にかかわらず，弾は移動する。

このうち，B.については前回の文字移動プログラムが縦になったものと考えればよいでしょう。とくに問題はないはずです。しかし，A.についてはどのようにして，プログラムを組めばよいのでしょう？

■フラグという考えかた

よく飲食店で「営業中」「準備中」といった札がドアに掛かっているのを見ることがあります。この札を見ることで，お客さんはお店が開店しているか，いないかを判断することができます。これがプログラムに応用できるのです。

まず，弾の状態を示す変数(これが札に当たります)を用意します。そして，弾が発射されたら（お店が開店したら）この変数に1を代入して弾が表示する状態であることを示します。また，弾が画面の外に移動したり，敵に当たったりしたら（お店が閉店したら）変数に0を代入して弾が表示しない状態であることを示します。

こうすることで，弾を表示するときに変数の値をチェックして，1であれば表示する，0であれば表示しないと判断できます。この考えかたをフラグと呼びます（**第３図**）。

これらをもとに作ったのが，**リスト3**のプログラムです。

新しく作成したのが，副プログラム

《リスト２》 基本的な副プログラムの使用例

```
10 CALL WRITE                    副プログラム「WRITE」を呼び出します。
20 SUB WRITE                     副プログラム「WRITE」ここから
30  PRINT "WRITE"
40 END SUB                       副プログラム「WRITE」ここまで。

自機(文字)移動プログラムの副プログラム版

100 DO
110  CALL P_MOVE                 副プログラム「P_MOVE」を呼び出します。(1000行に移動)
120  CALL WRITE                  副プログラム「WRITE」を呼び出します。(2000行に移動)
130  FSYNC 1
140 LOOP
1000 SUB P_MOVE                  副プログラム「P_MOVE」はここから
1010  IF PAD(1,1)>0 THEN PX=PX+1 自機の移動処理部分
1020  IF PAD(1,2)>0 THEN PX=PX-1
1030  IF PAD(1,3)>0 THEN PY=PY-1
1040 IF PAD(1,4)>0 THEN PY=PY-1
1050 IF PX<0 THEN PX=0
1060  IF PX>36 THEN PX=36
1070  IF PY<0 THEN PY=0
1080  IF PY>22 THEN PY=22
1090 END SUB                     副プログラム「P_MOVE」おわり。(120行に戻る)
2000 SUB WRITE                   副プログラム「WRITE」はここから
2010  CLEAR                      表示処理部分
2020  LOCATE PX,PY
2030  PRINT "A"
2040 END SUB                     副プログラム「WRITE」おわり。(130行に戻る)
```

※なお，ここでは「コントローラの状態から自機の座標を変化させる」部分と「自機の座表に文字を表示する」部分に分けて副プログラム化しています。

《リスト３》 フラグの考えを組み込んだプログラム・リスト

```
100 PX=18                        自機の初期位置設定
110 PY=22                        上に同じ
120 DO
130  CALL P_MOVE
140  CALL P_SHOT                 弾の発射 副プログラム呼び出し
150  CALL B_MOVE                 弾の移動 副プログラム呼び出し
160  CALL WRITE
170  FSYNC 1
180 LOOP
1000 SUB P_MOVE
1010  IF PAD(1,1)>0 THEN PX=PX+1
1020  IF PAD(1,2)>0 THEN PX=PX-1
1030  IF PAD(1,3)>0 THEN PY=PY-1
1040 IF PAD(1,4)>0 THEN PY=PY-1
1050 IF PX<0 THEN PX=0
1060  IF PX>36 THEN PX=36
1070  IF PY<0 THEN PY=0
1080  IF PY>22 THEN PY=22
1090 END SUB
2000 SUB WRITE
2010  CLEAR
2020  LOCATE PX,PY
2030  PRINT "A"
2040  IF B_FLAG=1 THEN           弾のフラグを見て表示できる状態ならば
2050   LOCATE B_X,B_Y            座標(B_X、B_Y)に
2060   PRINT "I"                 弾「I」を表示
2070  END IF
2080 END SUB
3000 SUB P_SHOT
3010  IF PAD(1,6)>0 THEN         ○ボタンが押されたら
3020   B_FLAG=1                  弾を表示可能状態にフラグを変更
3030   B_X=PX                    弾の座標を自機の前方に設定
3040   B_Y=PY-1
3050  END IF
3060 END SUB
4000 SUB B_MOVE
4010  IF B_FLAG=1 THEN           弾が表示可能状態ならば
4020   B_Y=B_Y-1                 弾を上に移動
4030   IF B_Y<0 THEN B_FLAG=0    弾が画面外に移動したらフラグを非表示状態に変更
4040  END IF
4050 END SUB
```

《リスト4》 配列を使って最大3発まで発射できるプログラム・リスト

```
100 B_MAX=3
110 DIM B_STORAGE(B_MAX),B_X(B_MAX),B_Y(B_MAX),B_FLAG
(B_MAX)
120 B_COUNT=1
130 FOR B_ID=1 TO B_MAX
140  B_STORAGE(B_ID)=B_ID
150 NEXT B_ID
160 PX=18
170 PY=22
180 DO
190 B_INTER=B_INTER-1
200 IF B_INTER<0 THEN B_INTER=0
210  CALL P_MOVE
220  CALL P_SHOT
230  CALL B_MOVE
240  CALL WRITE
250 FSYNC 4
260 LOOP
1000 SUB P_MOVE
1010  IF PAD(1,1)>0 THEN PX=PX+1
1020  IF PAD(1,2)>0 THEN PX=PX-1
1030  IF PAD(1,3)>0 THEN PY=PY-1
1040  IF PAD(1,4)>0 THEN PY=PY+1
1050  IF PX<0 THEN PX=0
1060  IF PX>36 THEN PX=36
1070  IF PY<0 THEN PY=0
1080  IF PY>22 THEN PY=22
1090 END SUB
2000 SUB WRITE
2010 CLEAR
2020  LOCATE PX,PY
2030  PRINT "A"
2040  FOR B_ID=1 TO B_MAX
2050   IF B_FLAG(B_ID)=1 THEN
2060    LOCATE B_X(B_ID),B_Y(B_ID)
2070    PRINT "I"
2080   END IF
2090  NEXT B_ID
2100 END SUB
3000 SUB P_SHOT
3010  IF PAD(1,6)>0 AND B_INTER=0 THEN
3020   IF B_COUNT>B_MAX THEN EXIT SUB
3030   B_ID=B_STORAGE(B_COUNT)
3040   B_FLAG(B_ID)=1
3050   B_X(B_ID)=PX
3060   B_Y(B_ID)=PY-1
3070   B_INTER=5
3080   B_COUNT=B_COUNT+1
3090  END IF
3100 END SUB
4000 SUB B_MOVE
4010  FOR B_ID=1 TO B_MAX
4020  IF B_FLAG(B_ID)=1 THEN
4030   B_Y(B_ID)=B_Y(B_ID)-1
4040  IF B_Y(B_ID)<0 THEN
4050   B_FLAG(B_ID)=0
4060   B_COUNT=B_COUNT-1
4070   B_STORAGE(B_COUNT)=B_ID
4080 END IF
4090  END IF
4100  NEXT B_ID
4110 END SUB
```

P_SHOT（弾発射部分）とB_MOVE(弾移動部分)です。弾が発射されるときと，移動して画面の外に出たときに，B_FLAGを切り替えているのがわかると思います。また，描画部分の副プログラムWRITEでは，B_FLAGを判断して弾の表示を行っています。

■実行して

さて，リスト3のプログラムを実行すると，弾の「I」が飛ぶようになりました。しかし,「ボタンを押したとき」ではなく「ボタンを離したとき」に弾が飛んだり，弾が飛んでいるときに再度○ボタンを押すと今まで飛んでいた弾が消えて新しく自機の前方から弾が発射してしまい，考えたとおりの動きをしてくれません。

これには二つの原因が考えられます。まず，一つ目は弾が発射されている間もつぎの弾が発射できるため，新しい弾が発射されると古い弾の座標は消されてしまうこと。二つ目はボタンが押されていれば，メインループがまわるたびに発射処理がされるために，ボタンを1回押したつもりでも実際には何度も弾を発射していることです。

配列を使う

まずは，弾を複数扱えるようにしながら，一定数の弾が画面上にある間は新しく弾が発射されないようにしましょう。弾を複数扱うには配列を使います。配列を使うことで，弾の処理が格段に楽になるからです。

先ほどの弾の移動では座標を指定するものとしてB_X，B_Yの二つの変数を使っていました。弾が一つしか発射しないのであればよいのですが，同じ物を複数処理する場合はそれぞれに対して別々の変数名を用意するのは手間が掛かってしまいます。ここで役に立つのが配列です。

配列は簡単にいうと変数の集まりのようなものです。変数が一つの変数名に一つの数値や文字列しか入れることができないのに対して，配列は配列名の横に書かれる数字で区別することで，複数の値を持つことができます(第4図)。

配列を使う前には変数とちがい，配列を使うことを宣言して，配列の用意をする必要があります。配列の宣言には「DIM」命令を使ってこのように書きます。

```
DIM 配列名（添字の最大数）
```

たとえば,

```
DIM A(10)
```

を実行することで「A」という配列が用意されました。括弧内の数字はAという配列の要素をいくつ用意するかの指示です。ここでは10を指定したので1～10までの10個の要素を持つ配列が用意されました。なお，配列名と変数名は同じ名前を使うことができません。

さて，配列の宣言が終わったら，実際に配列に数値を入れることができます。配列はつぎのように表記します。

```
配列名(添字)
```

配列名は「DIM」命令で宣言した配列名，その右側に括弧をつけて「添字」と呼ばれる識別用の数字を入れます。

実際に数値を代入すると，このようになります。

```
A(1)=10
A(2)=20
```

これで同じAという配列名でも添字を変えることでちがう値を入れられることがわかります。配列の利点は添字の部分に変数が入れられることです。これにより，複数の物に対して同じ処理をさせる場合に，添字の部分の変数の値を変更するだけで行うことができます。

配列を使って最大3発まで発射できるプログラムがリスト4です。

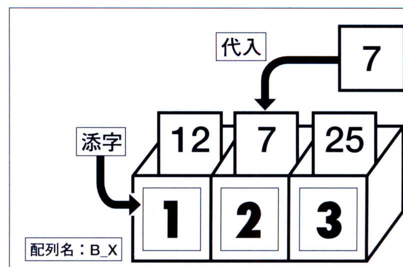
まず，100行で画面上に表示できる弾数の最大値を指定します。ここでは画面上に最大3発の弾が表示されるようにしています。

つぎに110行で弾に使用する配列の宣言を行っています。配列の内容は

B_X ……… 弾のX座標
B_Y ……… 弾のY座標
B_FLAG … 弾の状態フラグ（1：弾が画面上にある，0:弾が画面から消えている
B_STORAGE 弾のID格納用配列

となっています。

120行で初期値を入れている変数B_COUNTはB_STORAGEと組み合わせて発射する弾に管理番号を割り振るのに使います。なお，管理番号の初期値は130行から150行の間で入れています。



配列は，変数の箱がまとめられたようなもので，それぞれの箱に一つずつ，数値や文字列を入れることができます。また，箱には添字と呼ばれる識別用の数字がつけられており，配列名と添字を指定することで，一つの箱を指定します。

《第4図》 配列とは何か

弾が発射されるときは3030行で弾に管理番号を発行しています。また，画面の外に移動した弾の管理番号は4070行で配列，B_STORAGEに返却しています。

また，3010行では発射ボタンが押されているかどうかの判断と同時にカウンタB_INTERが0であるかどうかのチェックを入れています。B_INTERは弾が発射されると5が代入され，メインループが1回まわるごとに1ずつ減少させています。これにより，一度弾が発射されたら，つぎの弾が発射されるまでの間隔を取っているのです。

■FOR～NEXT文

さて，ここでまた，新しい命令が出てきました。FOR～NEXT文です。これは特定の回数処理を繰り返すのに便利な命令で，つぎのような書きかたをします。

FOR（カウンタ用の変数の定義）TO（カウンタ用の変数がいくつになるまで繰り返すか）
**　　繰り返す処理部分**
**　NEXT （カウンタ用の変数）**

このように書くと，FORの後に書かれたカウンタ用の変数が繰り返し部分の処理を行うごとに1ずつ加算されていき，繰り返し回数に達した時点で繰り返しを終了します。また，カウンタ用の変数はそのまま変数として使えます。ここではカウンタ用の変数を弾の座標用の配列に添字として使うことで，1回目のループは1番目の弾を，2回目のループは2番目の弾といったように弾を順番に処理しています。

■弾の管理

さらに，複数の弾を管理するために，B_COUNTとB_IDという二つの変数と，B_STORAGEという配列を使用しています。これらの関係は少し複雑ですが，**第5図**を見ればどのように管理されているかが理解できると思います。

このような複雑な管理をしているのは，弾が発射された順番に消えるとは限らないため，常に画面上に表示されていない(空いている)弾の番号を把握しておく必要があるからです。

■敵の登場と処理

さて，これまでの説明で，ひととおり自機に関する処理が終わりました。つぎは敵の登場と移動部分の作成に入ります。ここでは先ほどの「弾の発射処理」をうまく流用して作ってみることにします。

敵の出現と移動の要素はこのような感じとなります。

①画面上部から出現(横の位置はランダム)
②出現したら，画面下に向かって移動する。

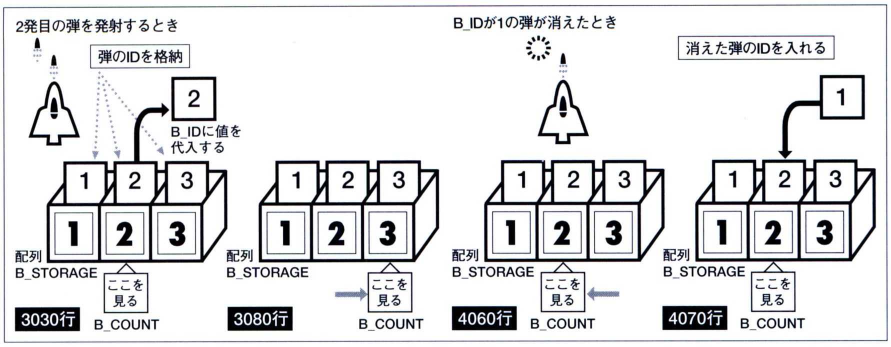
これは，先ほどの弾の発射処理から「○ボタンが押されたら発射」の部分を除いて，出現座標を自機の前方から画面上部に変更し，移動方向を逆にすれば簡単に作ることができます（リスト5）。

基本的な敵の登場と移動の処理は弾の発射処理と同じなのでわかると思います。

なお，敵が出現する際の横座標を決定するために「RND」関数を使用しています。RND関数は0から1未満の数字をランダムに返してきます。試しに

```
PRINT RND
```

と実行すると，実行するたびにちがう数字が出てきます。5050行では，このRNDに36を掛けて0～36までの数字をランダムに発生させています。発生させた横座標は，関数の説明に使用したIP関数を使って整数に変



《第5図》 配列による弾の管理

《リスト5》 出現座標を画面上部に変更し，移動方向を逆にしたプログラム・リスト

```
100 B_MAX=3!                    ―画面内の自機の弾数
110 E_MAX=5!                    ―画面内の敵の数
120 DIM B_STORAGE(B_MAX),B_X(B_MAX),B_Y(B_MAX),B_FLA
G(B_MAX)
130 DIM E_STORAGE(E_MAX),E_X(E_MAX),E_Y(E_MAX),E_FLA
G(E_MAX)
140 B_COUNT=1
150 FOR B_ID=1 TO B_MAX
160  B_STORAGE(B_ID)=B_ID
170 NEXT B_ID
180 E_COUNT=1
190 FOR E_ID=1 TO E_MAX
200  E_STORAGE(E_ID)=E_ID
210 NEXT E_ID
220 PX=18
230 PY=21
240 DO
250    CALL P_MOVE
260   CALL P_SHOT
270  CALL B_MOVE
280   CALL E_ENTRY
290   CALL E_MOVE
300  SHOT_INTER=SHOT_INTER-1
310  IF SHOT_INTER<0 THEN SHOT_INTER=0
320  E_INTER=E_INTER-1
330  IF E_INTER<0 THEN E_INTER=0
340  FSYNC 4
350  CALL WRITE
360 LOOP
1000 SUB P_MOVE
1010  IF PAD(1,1)>0 THEN PX=PX+1
1020  IF PAD(1,2)>0 THEN PX=PX-1
1030  IF PAD(1,3)>0 THEN PY=PY-1
1040  IF PAD(1,4)>0 THEN PY=PY+1
1050  IF PX<0 THEN PX=0
1060  IF PX>36 THEN PX=36
1070  IF PY<0 THEN PY=0
1080  IF PY>21 THEN PY=21
1090 END SUB
2000 SUB WRITE
2010  CLEAR
2020  LOCATE PX,PY
2030  PRINT "A"
2040  FOR B_ID=1 TO B_MAX
2050   IF B_FLAG(B_ID)=1 THEN
2060    LOCATE B_X(B_ID),B_Y(B_ID)
2070    PRINT "I"
2080   END IF
2090  NEXT B_ID
2100  FOR E_ID=1 TO E_MAX
2110   IF E_FLAG(E_ID)=1 THEN
2120    LOCATE E_X(E_ID),E_Y(E_ID)
2130    PRINT "V"
2140   END IF
2150  NEXT E_ID
2160 END SUB
3000 SUB P_SHOT
3010  IF PAD(1,6)>0 AND SHOT_INTER=0 THEN
3020   IF B_COUNT>B_MAX THEN EXIT SUB
3030   B_ID=B_STORAGE(B_COUNT)
3040   B_FLAG(B_ID)=1
3050   B_X(B_ID)=PX
3060   B_Y(B_ID)=PY
3070   SHOT_INTER=5
3080   B_COUNT=B_COUNT+1
3090  END IF
3100 END SUB
4000 SUB B_MOVE
4010  FOR B_ID=1 TO B_MAX
4020   IF B_FLAG(B_ID)=1 THEN
4030    B_Y(B_ID)=B_Y(B_ID)-1
4040    IF B_Y(B_ID)<0 THEN
4050     B_FLAG(B_ID)=0
4060     B_COUNT=B_COUNT-1
4070     B_STORAGE(B_COUNT)=B_ID
4080    END IF
4090   END IF
4100  NEXT B_ID
4110 END SUB
5000 SUB E_ENTRY
5010  IF E_INTER=0 THEN
5020   IF E_COUNT>E_MAX THEN EXIT SUB
5030   E_ID=E_STORAGE(E_COUNT)
5040   E_FLAG(E_ID)=1
5050   E_X(E_ID)=IP(RND*36)
5060   E_Y(E_ID)=0
5070   E_INTER=5
5080   E_COUNT=E_COUNT+1
5090  END IF
5100 END SUB
6000 SUB E_MOVE
6010  FOR E_ID=1 TO E_MAX
6020   IF E_FLAG(E_ID)=1 THEN
6030    E_Y(E_ID)=E_Y(E_ID)+1
6040    IF E_Y(E_ID)>21 THEN
6050     E_FLAG(E_ID)=0
6060     E_COUNT=E_COUNT-1
6070     E_STORAGE(E_COUNT)=E_ID
6080    END IF
6090   END IF
6100  NEXT E_ID
6110 END SUB
```

化させています。

当たり判定

敵が出現したことで，かなりゲームらしくなってきました。しかし，今のままでは弾を発射しても敵を素通りするだけです。当たり判定を作成して，敵に弾が当たるようにしてみましょう。

さて，例によって，敵に弾が当たったときの要素を考えてみることにします。

①弾と敵が重なったら，当たったことにする
②当たったら，弾は消える
③当たったら，敵は爆発する

①は敵が重なっている＝弾と敵の座標が一致していると考えることができます。これを使って当たり判定をチェックしてみましょう。
②は弾が画面の外に消える時の処理を流用します。
③は敵の状態フラグを変化させて，それに対応して爆発文字を表示させることにします。

この要素を踏まえて作成したのが**リスト6**です。

7000行から7180行までが弾と敵が当たっているかをチェックする部分です。FOR～NEXT文が入れ子構造になっているので少し複雑ですが，これによってすべての弾と敵についてチェックしています。

実際に重なっているかどうかの判断は7030行のIF文で判定しています。少し複雑なのは弾と敵のX座標，Y座標のチェックの他に弾と敵が現在画面上にあるかどうかもチェックしているからです。これは，弾や敵が画面外に移動して消えたりしても，座標自体は残っているからです。

7040行から7060行までは弾が敵に当たった際に弾を消している処理です。これは，画面の外に弾が移動したときと同じように処理しています。

つぎに，7070行から7080行で敵はまだ消さないでフラグ「E_FLAG()」を変化させて爆発状態としています。爆発状態となった敵は，描画を行う副プログラムWRITEでフラグがチェックされ，2190行から2210行で爆発用の文字として「○」と「＊」が書かれます。

なお，爆発した敵が一定時間で消えるように，爆発用のカウンタ「E_EXPTIME()」を設けています。敵が爆発している間，カウンタは減少し，カウンタが0になった時点で初めて敵は消去されます。

同様にして，自機と敵との衝突判定も作成します。自機は一つしかないので，弾と敵のあたり判定よりは楽に作成できると思います。

細かい仕様を付け加える

これでゲームとしての大まかな部分が完成しました。後は，得点やゲーム・オーバーの処理を入れれば，ゲームらしくなります。このあたりの細かい処理は，今までのプログラムの応用でできると思います。**リスト7**にテキスト・バージョンでの完成リストを掲載しますので，打ち込んでみてください。

なお，ここでは自機，自機の弾，敵それぞれに移動速度のちがいを持たせるため，少し工夫をしています。メインループでカウンタとして変数「CNT」を1ずつ加算して，自機の移動処理はメインループが2回につき1回，敵の移動処理は3回につき1回処理することで，それぞれちがいをつけています(自機の弾は毎ループ処理されます)。

ここで条件文に使用している関数「REMAINDER(A,B)」はAをBで割った際の余りが値として返ってくる関数です。よって360行の条件文では，REMAINDER(CNT,2)を使うことで，カウンタ「CNT」の現在値が2で割り切れるかどうかを判断しているのです。

また，550行ではカウンタ「CNT」を6で割った際の余りを再度「CNT」に代入することで，CNTを0～5の間でループさせています。

最後に

今回は，テキスト文字を使ってシューティング・ゲームを作成してみました。前回に比べて，やや詰め込みすぎた感じもありますが，どれも覚えておいて損のない命令や考えかたです。1回読んでみて理解できない場合は，もう一度読んでみたり，プログラムを実際に打ち込んで動かしてみてください。

少しずつ，理解していきましょう。また，プログラム中にPRINT文で配列や変数を表示すると，配列や変数の値の変化が目にみえてわかるのでお勧めです。詰まってみたらやってみましょう。

次回は，3Dモデルの扱いかたを学びながら，今回作成したテキスト版のシューティング・ゲームを3Dモデルを使ったものに改造します。楽しみにしてください。

《第1表》 配列と変数の一覧表

	名前	役割
自機関連	PX	自機のX座標
	PY	自機のY座標
	P_FLAG	自機の状態フラグ 1：通常 2：爆発 3：無敵（復活時）
	P_EXPTIME	自機の爆発時間用カウンタ
	MUTEKI_TIME	自機の無敵時間用カウンタ
弾関連	B_MAX	弾の最大数
	B_STORAGE	弾の管理番号格納用配列
	B_X	弾のX座標
	B_Y	弾のY座標
	B_FLAG	弾の状態フラグ 1：弾画面内にあり 2：弾画面内になし
	B_COUNT	B_STORAGEと組み合わせて弾の管理番号の振り分けに使う
	B_ID	弾の管理番号
	B_INTER	弾の発射間隔調整用カウンタ
敵関連	E_MAX	敵の最大数
	E_STORAGE	敵の管理番号格納用配列
	E_X	敵のX座標
	E_Y	敵のY座標
	E_FLAG	敵の状態フラグ 1：敵画面内にあり 2：敵画面内にいない 3：爆発状態
	E_COUNT	E_STORAGEと組み合わせて敵の管理番号の振り分けに使う
	E_ID	敵の管理番号
	E_EXPTIME	敵の爆発時間
	E_INTER	敵の登場間隔調整用カウンタ
その他	MISS	自機がやられた回数
	SCORE	得点

「夏休み公開プログラミング講座」有意義に行われる！

主催：ベーマガ編集部
協賛：デジタルエンタテインメントアカデミー／アートディンク

プログラムの楽しさを多くのみなさんに体験してもらおうと企画した“夏休み公開プログラミング講座”が7月30日（月）と31日（火）にデジタルエンタテインメントアカデミーの校舎で行われました。

当初の募集人員をあっという間にオーバーしたため，急きょ，2日間の開催に切り替え，この特集の筆者でもあるアートディンクの齋藤博希氏に講師になってもらい，「BASIC STUDIO」を使用した，簡単な3Dシューティング・ゲームの制作を体験してもらいました。参加したほとんどの人は初心者でしたが，プログラムに興味があるという人ばかりで，熱心に受講していました。



パソコンレクチャー ザ・ビギニング

by
くりひろし

Step1: 「驚異の指コンピュータ!!」



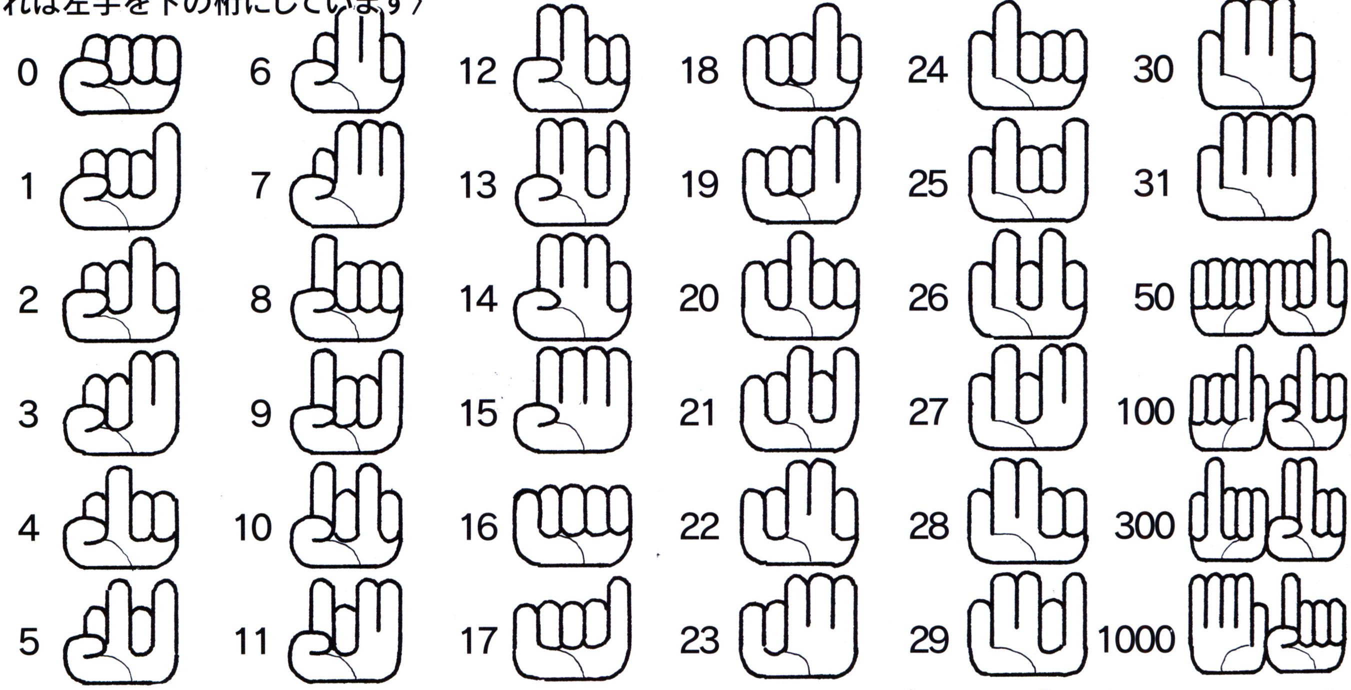

ミーン
ミーン
…と, いうわけで次号の記事に関連した調査を…
…ムムッ
カッコつけてテラス席なんかにいるから…
あたしのPC〜
ハハッ今ごろ正気にもどったな
なーにパソコンの1台や2台1日2日なくたって仕事くらい…
キミがソレ言うんじゃないよう〜
んなワキャないのよう困ってるのよう
じたばた
20年前ならねェ
んーだってほんの10年前はノートPCだってなかったんだよぉ？
そもそも「パソコン」なんてなかったころからベーマガはあったんだし…みんなパソコンに生活たよりすぎだよ
オオッ影くんがマトモなコトを…
今, 身近な製品にはコンピュータが組み込まれていて, それが当たり前に思われているけど…
ベーマガ創刊のころなんておよそ大したものなんてなかったんだよ
Pentium4 1.8GHz
iモードケータイ
デジカメ
液晶ディスプレイ
DVD-RW付タワーPC
うーむ
Z80
プッシュホン
35mm銀塩
モノクロCRT
データカセット付き
ここらで一度パソコンって何か見なおしてもいいんじゃないかな──タダの便利な魔法の素じゃなくてさ!!
ネ…ネツないかー!?
そーじゃのう! どうしてゲームができたりインターネットができたりするのか？そもそもパソコンってどうやって動いているのか？
基本中の基本──まさしくパソコンのベーシックなところを見なおしてみよう!
まずはパソコンって何!? ですが…
コンピュータ
COMPUTER
▶電子計算機
▶電子頭脳
まァパーソナルってのはわかるけどね。
パソコン=パーソナル・コンピュータ
Personal Computer
(個人用のコンピュータ)
うむ,「パソコン」を理解するには「コンピュータ」を理解しなければならんじゃろうな



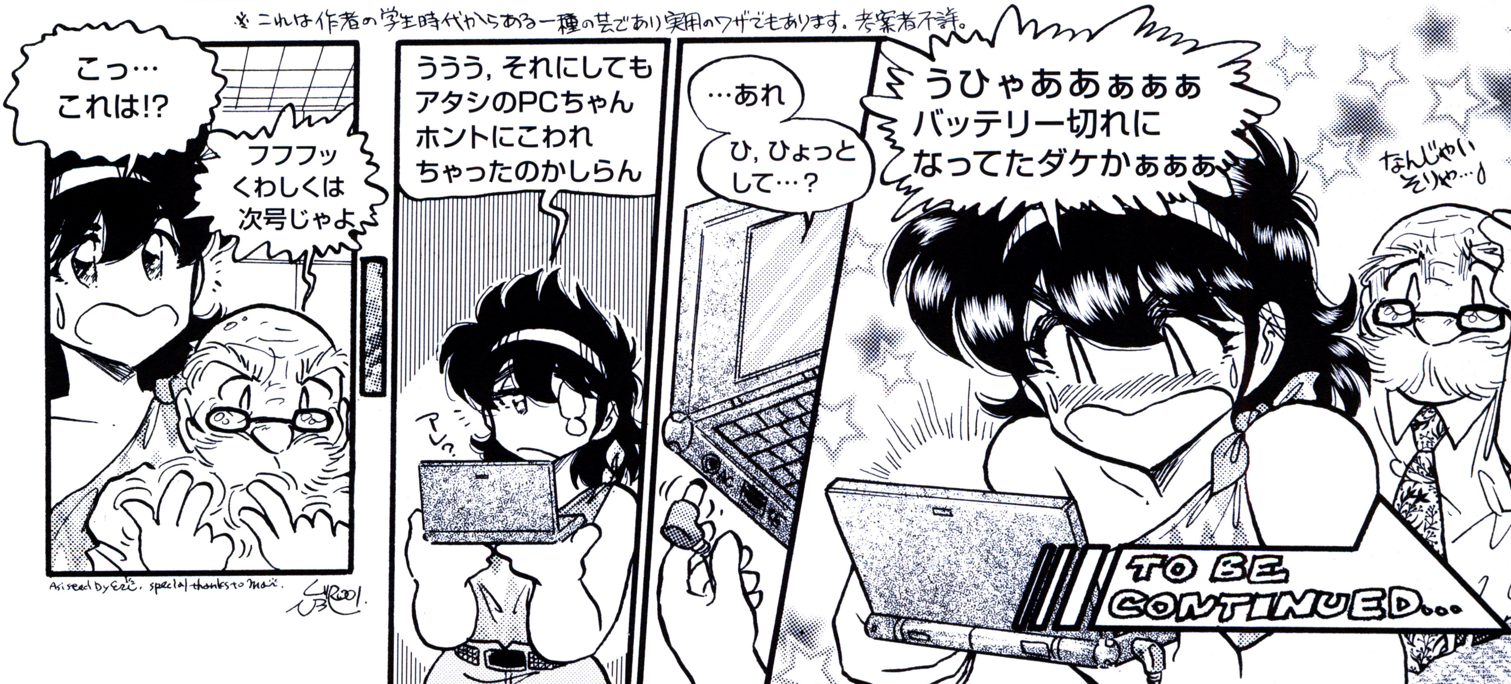
コンピュータの歴史なんぞたかだか100年もないパソコンなんて10年そこそこだ
基本的な動作原理や構成は現在にいたるまで大きく変わるものではないのだ
えー，変わってない!?…だってゼンゼン形とか…
性能！集積度(どれだけ小さくたくさんつめこめるか！)はたしかに進歩した！
しかし！見かけがちがっても T型フォードとフェラーリは同じ車であるのと同じ！
絶対的に同じことがひとつ!!
コンピュータは電気で動くのである!!
お？
ヒソ ヒソ
アタリマエよねエー
うぬうーっバカにしとるなァーッ「何で」電気で動くのかもわかっとらんくせに〜〜
はあいっ どーして電気で動くのかわかるヒトっ
はあいっ ミニ四駆モータで動きますッ
MOTOR
影さんの出番はココまでだったりする
んー…モールス信号みたいについたり消えたりの長さ…かなァ？
Dr.Dの…ひとくちメモ？
えーっと最近はあまり使っておらんかもしれんが…えー「無線」通信のじゃな最もキホン的でかつトラディッショナルな方式のひとつに，信号音の区切る長さの組み合わせで意味を伝えるものがあります。コレがモールス信号とゆーものです。
B—… A・— S… I・・ C—・—・
ワシって古い？
いやいや，しかし！それはスゴク近い!!
あら
コンピュータの動作原理の基本は電気(による信号)が「ある」か，「ない」かをすなわち「1」か「0」かでおきかえたものなのである!!
ON
OFF
1
0
ある
電流が
ない



〈これは左手を下の桁にしています〉



※これは作者の学生時代からある一種の芸であり実用のワザでもあります。考案者不詳。



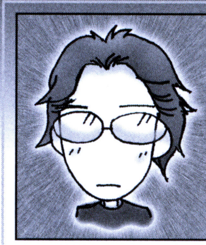
Technical News Spot

速度が上がるとパソコンの使いかたが劇的に変わる！

テクスポ　2001年9月号

x86互換CPUメーカーであるインテル，AMD，トランスメタ，VIAから，続々と新製品が発表されています。今月は，AMDの新しいCPUについて紹介しつつ，『JBuilder 5』の続報と，マイクロソフト・ドットネットの続報をお届けします。

TEXT●石川直人



最近1年間で，10kg太りました。現在の体重78kg。さすがにやばいです。というわけで，今月からダイエットをはじめることにしました。毎月この欄で経過を報告しますので，なま暖かく見守ってください。

ニュース

PalominoコアのAthlon MP登場

AMDの新CPU『Athlon MP』が発表されました。

“Palomino”というコードネームで呼ばれていたCPUシリーズとしては，すでに発表されている『Mobile Athlon 4』につぐ製品です。1GHz版と1.2GHz版が発表されており，すでに秋葉原等で購入することができます。

Palominoシリーズの特徴としては，以下のようなものがあります。

●ハードウエア・プリフェッチ機能付きキャッシュ

従来のキャッシュでは，CPUが一度アクセスしたメモリの内容を高速なキャッシュ・メモリに保存しておき，それを再利用する場合にのみ，遅いメインメモリではなく高速なキャッシュ・メモリにアクセスしていました。

Palominoで採用された新しいキャッシュ・システムは，必要となりそうなデータ（メモリ領域）を予測し，それをあらかじめキャッシュ・メモリに取り込んでしまうことで，従来以上にメモリアクセスを高速化しています（**第1図**）。

●3DNow! Professional

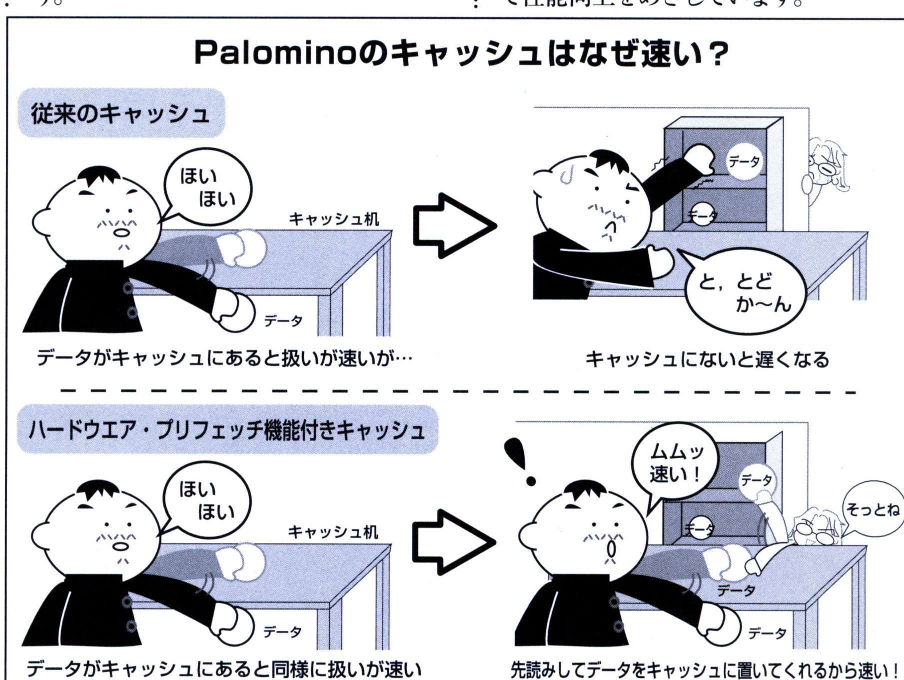
マルチメディア機能を高速化する独自命令セット“3DNow!”が新しくなりました。

この3DNow! Professionalは，従来のEnhanced 3DNow!との互換性を保ったまま，インテルの『Pentium Ⅲ』以降に搭載されているマルチメディア命令セットSSE互換の命令が追加されています。これにより，SSE対応アプリケーションの実行速度が高速化されることが期待できます。

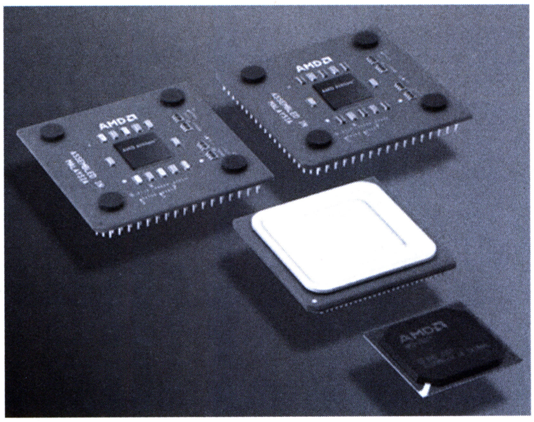
もともとAMDのAthlonは，同クロックのPentium Ⅲ／Pentium 4よりも性能が高いという評判でした。おそらく今回搭載された新機能によって，さらにクロックあたりの性能が向上することでしょう。

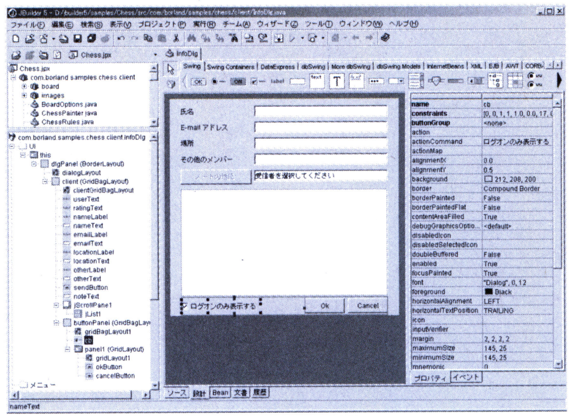
一方のインテルは，Pentium 4のクロックをさらに引き上げ，現在（2001年7月下旬）のところ，最速クロック1.8GHzの製品を発表しています。

どちらかというと，クロックあたりの性能向上を中心に高速化を図るAMDに対して，大幅なクロック向上によって高速化を図るインテル。また，トランスメタは根本的に設計思想の異なる『Crusoe』で性能向上をめざしています。



《第1図》ハードウエア・プリフェッチ機能付きキャッシュはこんな風にエライ！（どこかで見たような2人？）


《写真1》これがAthlon MPだ。秋葉原では，30,000円ほどで販売されている（2001年7月下旬）


《写真2》JBuilder 5 Personal版の画面。非商用ならば，無料で使用することができるのがうれしい

最近のCPU開発競争は，各社それぞれ独自の技術を駆使しており，なかなか興味深い様相となっているようです。

ニュース

JBuilder 5の続報

先月紹介したボーランドのJava開発ツールの最新版『JBuilder 5』が，正式に発表されました。日本での出荷開始日は，7月12日。すでに使われている人もいるでしょうか。製品形態としては，Personal（パーソナル）／Professional（プロフェッショナル）／Enterprise（エンタープライズ）の3パッケージが用意されます（**第1表**）。

前バージョンまでオンラインで無料配布されていたFoundation（ファウンデーション）バージョンがラインナップからなくなっていますが，これは単に新しくPersonalバージョンと名称を変更しただけのようです。

Personalバージョンには，CD-ROM提供（マニュアル付き）で4,800円という有償販売と，インターネット上での無償ダウンロードの2とおりの配布形態がサポートされるそうです。

Windows／Linux／Solaris（そして将来的にはMac OS X）とマルチ・プラットフォームに対応したJBuilder 5と，同じくマルチ・プラットフォームに対応したJavaの組み合わせで，プログラミングの世界を広げてみてはいかがでしょうか？

ニュース

ドットネットがFreeBSDで動作する

マイクロソフトはコーレル社と協力し，ドットネット・プラットフォームをWindows環境だけでなく，UNIX環境にも移植することを発表しました。

ただし，対象となるOSはかねてからウワサになっていたLinuxではなく，“FreeBSD”（Linux同様のフリーUNIX系OS）となりました。また，サポート内容はあくまで研究などの非営利目的に限られた，制限付き「共有ソースコード」計画の一環とされるそうです。

具体的にFreeBSDでサポートされるドットネットの機能は，C#言語とCLI（Common Language Infrastructure）（コモン ランゲージ インフラストラクチャ）となるようです。

C#は，マイクロソフトがJava対抗として新しく開発した新言語です。CLIは，JavaにおけるJava仮想マシンのように，複数のシステム上でドットネット・アプリケーション（C#プログラムなど）を動作させるための仕組みです。

今後の新しい標準環境になりうる可能性を秘めたドットネットが，Windows以外のOSにも移植されるという事実は，非常に興味深いものです。マイクロソフト，およびコーレルの今後の動向に注目しましょう。

《第1表》JBuilder 5のバージョンと価格

製品名	Borland JBuilder 5 Personal	Borland JBuilder 5 Professional	Borland JBuilder 5 Enterprise
希望小売価格	4,800円*1	68,000円	360,000円
動作OS	Windows 98／2000／NT4，Linux，Solaris		

*1　無償ダウンロードも可能。

今月の予想

オープンなはずのドットネット時代でもマイクロソフトの独占状態は変わらない！

「チップの性能は18か月で2倍になる」というムーアの法則は，30年以上前にインテルの共同創設者ゴードン・ムーアが提唱して以来，成立し続けています。現在，半導体レベルでの新技術が続々と発表されており，今後もムーアの法則は安泰であるといわれています。

また，最近ではコンピュータの性能のもう一つの指標となりつつある，インターネットへの接続速度もどんどん向上しています。あのYahoo!（ヤフー）が低価格なADSL接続サービス『Yahoo! BB』を発表するなど，k（キロ）ビット／秒単位があたりまえだった接続速度が，今や一般家庭でもM（メガ）ビット／秒が珍しくなくなってきています。

そんな中で，コンピュータ・プラットフォームをネットワーク中心に移行しようとするマイクロソフトのドットネットのこころみも，だんだんと具体性を帯びてきました。強力なCPUパワーと高速ネット回線によって，コンピュータの使いかたが劇的に変化する下地はもう十分にできているのです。

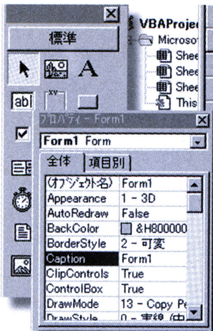
ただし，今回のマイクロソフトとコーレルによる，ドットネットのFreeBSDへの移植という話は，あまりにも制約が大きすぎ，本気ですべてのプラットフォームにドットネットを普及させる気はないように見えます。

おそらく今回のドットネットのFreeBSDへの移植という話は，「ドットネットは広いプラットホームをサポートしますよ」というマイクロソフトのポーズにすぎないのではないでしょうか？

結局のところ，ドットネット時代になっても，パソコン業界に対するマイクロソフトの（ドットネット・サーバによる）独占状態は変わらないものと予想します。

ドットネットは，もうちょっとオープンな規格になるのではないかと期待していたんですけどねぇ…。





はじめよう Microsoft Excelでの BASICプログラミング

協力 富士通南九州システムエンジニアリング

第5回 toto予想プログラムにチャレンジ（1）

この連載では，Microsoft Excelを使ってBASICのプログラミングを行います。今月は，簡単なシミュレーション・プログラムを作成します。

TEXT & PROGRAMING 椛島たけし



夏はビールがおいしい。猛暑でさらにおいしい。がぶがぶ飲んでいるが，ジーンズやスーツのウェストが苦しくなってきている。DirectXでウェストをトランスフォームせねば？

予想プログラム

結果の予想プログラムというと複雑なアルゴリズムを連想します。シミュレーションの場合，膨大なデータを，環境などの要因を考慮し，時間ごとの処理をするため大掛かりなものとなりそうです。しかし，元となるデータが多くなく，時間ごとの処理がない場合は，意外に簡単にシミュレーション（結果予想）を行うことができます。

toto予想プログラム

“toto”ことサッカーくじが発売されました。19歳未満は買うことができませんのでせめて予想だけでも楽しみましょう。ということで，サッカーの勝敗予想を行うプログラムを作成することにします。サッカーなどのスポーツの結果を予想する場合，どのようなことを考えますか？大体，つぎのようなことを想像すると思います。

「いい選手がいる」
「エースが故障した」
「最近調子がいい」
「現在順位が上／下」
「あのチームとは相性がいい」
「ホームゲームだ」

このような結果を左右する情報を「要素」として区分してみると，大体つぎのようなものに分類されるのではないでしょうか。

「能力」「調子」「実績」「相性」「環境」

これらの要素をうまく点数化(数値化)し，合計点数を求めることができれば，結果の予想が可能となりそうです(第1図)。

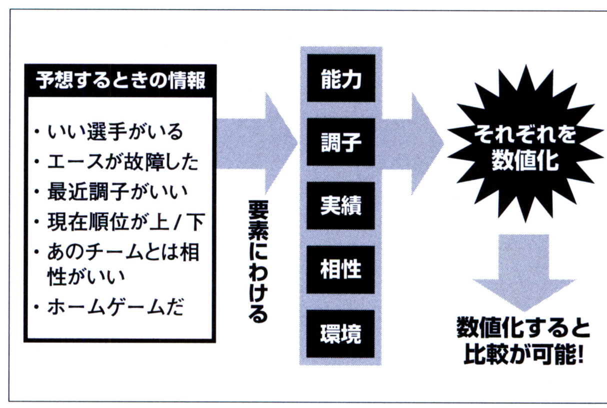
要素を数値化する方法

要素を数値化する方法について，どのようにすればよいのか考えてみましょう。

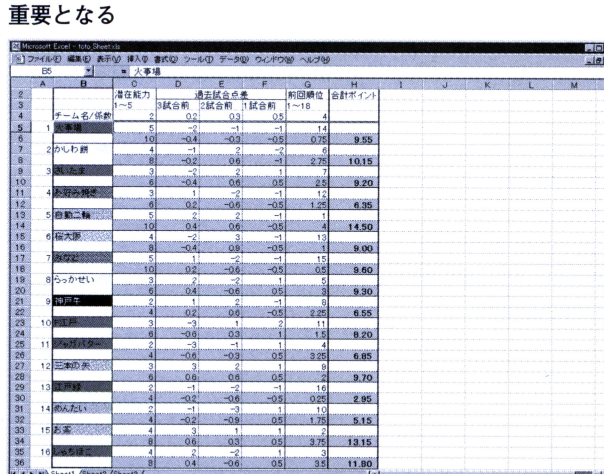
まず，「能力」については，デジタルに判断することがむずかしい分野ですね。ここは，「いい選手/監督がいる」とか「選手が出場停止」などの要素を5段階程度に自己判断してみましょう。わからなければ，「なんとなく強そう/弱そう」で判断してもOKです。

続いて「調子」については最近3試合程度の得失点差を利用することにします。「実績」については現在順位が利用できそうです。「相性」については，プログラムがちょっと複雑になりますので今回は省くとしましょう。

「環境」はいろいろありますが，気象などは基本的なものです。今回はホーム（地元）チームを有利にするように考えてみましょう。

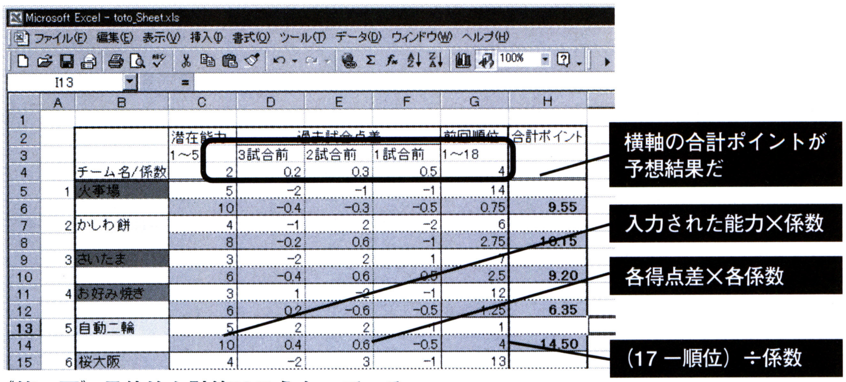


《第1図》シミュレーションには結果を左右する要素の数値化が重要となる

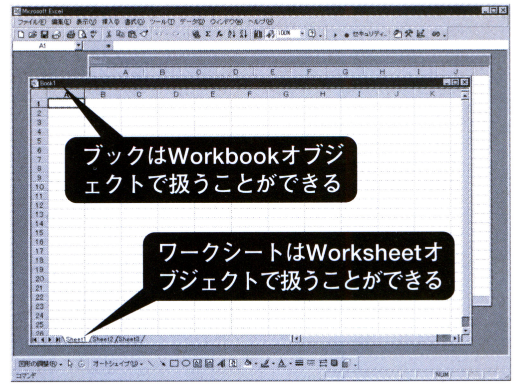


	A	B	C	D	E	F	G	H
2			潜在能力	過去試合点差			前回順位	合計ポイント
3			1〜5	3試合前	2試合前	1試合前	1〜18	
4		チーム名/係数	2	0.2	0.3	0.5	4	
5	1	火事場	5	−2	−1	−1	14	
6			10	−0.4	−0.3	−0.5	0.75	9.55
7	2	かしわ餅	4	−1	2	−2	6	
8			8	−0.2	0.6	−1	2.75	10.15
9	3	さいたま	3	−2	2	1	7	
10			6	−0.4	0.6	0.5	2.5	9.20
11	4	お好み焼き	3	1	−2	−1	12	
12			6	0.2	−0.6	−0.5	1.25	6.35
13	5	自動二輪	5	2	2	−1	1	
14			10	0.4	0.6	−0.5	4	14.50
15	6	桜大阪	4	−2	3	−1	13	
16			8	−0.4	0.9	−0.5	1	9.00
17	7	みなと	5	1	−2	−1	15	
18			10	0.2	−0.6	−0.5	0.5	9.60
19	8	らっかせい	3	2	−2	1	5	
20			6	0.4	−0.6	0.5	3	9.30
21	9	神戸牛	2	1	2	−1	8	
22			4	0.2	0.6	−0.5	2.25	6.55
23	10	F江戸	3	−3	1	2	11	
24			6	−0.6	0.3	1	1.5	8.20
25	11	ジャガバター	2	−3	−1	1	4	
26			4	−0.6	−0.3	0.5	3.25	6.85
27	12	三本の矢	3	3	2	1	9	
28			6	0.6	0.6	0.5	2	9.70
29	13	江戸好	2	−1	−2	−1	16	
30			4	−0.2	−0.6	−0.5	0.25	2.95
31	14	めんたい	2	−1	−3	1	10	
32			4	−0.2	−0.9	0.5	1.75	5.15
33	15	お茶	4	3	1	1	2	
34			8	0.6	0.3	0.5	3.75	13.15
35	16	しゃちほこ	4	2	−2	1	3	
36			8	0.4	−0.6	0.5	3.5	11.80

《第2図》Excelのシートで要素を数値化する



《第3図》具体的な計算はこうなっている



《第4図》ブックやワークシートはVBAからオブジェクトとして扱うことができる

Excelのセルでやってみる

まずはExcelのワークシートを使用して作成してみることにします。Excelは計算が得意ですので，簡単に一覧表を作成することができますね(第2図)。事前準備として，つぎのことを行います。

・5段階の「潜在能力」を入力します

有力な選手が故障したり，出場停止者がいる場合は減らします。有望な選手が移籍したりした場合は増やします。

・過去3試合の得失点差を入力する

スポーツ新聞に載っています。勝った場合はプラス，負けた場合はマイナスの値となります。

・前回順位を入力する

これもスポーツ新聞に載っていますね。

入力された値をそのまま足してもおもしろくないので，係数を決めておき，掛けたり，割ったりして，より現実的(?)な数値に近づけます。係数とその計算式が予想の心臓部となります。筆者はつぎのような計算式を考えてみました(第3図)。

[潜在能力]

そのまま，係数を積算しています。初期値は2なので単純に2倍しています。

・計算式→潜在能力×係数

[過去試合点差]

3試合前，2試合前，1試合前ごとに，係数を用意し積算しています。最近の試合ほど係数が大きくしています。値が大きくなるほど調子を重視しています。

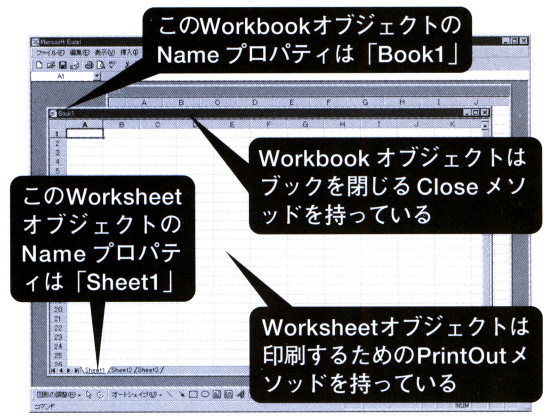
・計算式→3試合前点差×3試合前用係数+2試合前点差×2試合前用係数+1試合前点差×1試合前用係数

[前回順位]

17から引いて係数で割っています。17から順位を引くのは，16チームのため，最低が1，最高が16とするためです。また，値が大きすぎますので指定した係数で割ることにしています。したがって，係数が大きいほど実績を重視しないことになります。

・計算式→(17－前回順位)÷順位係数

これらの係数を変えることで，いろいろな角度からの結果予想が可能となります。これだけでも結構予想の足しにはなるかもしれませんね。しかし，ここでいくつか問題があります。対戦チームごとに合計ポイントをみるのが面倒です。また，係数を設定すのも面倒ですね。さらに，このままの一覧表では，ホームチームを有利にするのがむずかしそうです。そこで，Excel VBAを使用して，対戦チームの選択や係数，ホームチームの設定など楽にしまてみましょう。



《第5図》オブジェクトはいろいろなプロパティやメソッドを持っている

Excel VBAでワークシートのデータを取り出すには？

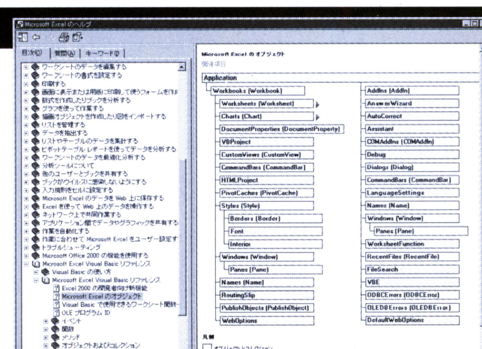
ワークシートやセルはオブジェクトとして扱うことができます。オブジェクト(!?)と聞いて難しそうに感じるかもしれませんね。コントロールのようなもので，プロパティやメソッドを持っています。コントロールとの大きなちがいは，オブジェクトの中に別のオブジェクトを持つことができる点です。

Excelの場合，ブックという単位でファイルを開きます。これがWorkbookオブジェクトになります。また，シートと

Techinical Column

Excelが持つオブジェクト

オンラインヘルプの「Microsoft Excel Visual Basicリファレンス」の「Microsoft Excelのオブジェクト」を一度参照してみよう。非常にたくさんのオブジェクトが掲載されている。


Excelには多くのオブジェクトがある。マニュアルで確認しておこう

いう単位でデータを扱います。これがWorksheetオブジェクトとなります(第4図)。

WorkBookオブジェクトやWorksheetオブジェクトは，当然プロパティやメソッドを持ってます(第5図)。そして，Workbookオブジェクトの中にはWorkSheetオブジェクトが含まれています（第6図)。

コレクションとは？

コレクションとはオブジェクトが格納されている配列のようなものです。たとえば，WorkSheetsコレクションにはWorkSheetオブジェクトが格納されています。Excelのシートは一つのブックに複数設定することができます。これら複数のシートをまとめて表しているのがWorkSheetsコレクションというわけです（第7図)。

コレクションに格納されているオブジェクトを使用する場合，コレクション名にインデックス番号やオブジェクト名を指定します。

たとえば，最初のシートを表すには，

```
WorkSheets（1）
```

と指定します。
シート名が「Sheet1」の場合

```
WorkSheets（"Sheet1"）
```

とすることもできます。配列の要素に数字だけでなく，識別名も使えるような感じですね。コレクションにインデックスが指定され，オブジェクトが特定されると，オブジェクトそのものを指します。つまり，WorkSheets(1)は，WorkSheetオブジェクトを表すのです。したがって，WorkSheetオブジェクトのプロパティやメソッドを使用することができます。

コレクションを使用すると，ブック上のすべてのシートに操作を行うときや，シート名がわからないときなどに便利です。

オブジェクトを返すプロパティ？

データはシート上のセルに格納されています。セルのデータはRangeオブジェクトのValueプロパティで参照できます。

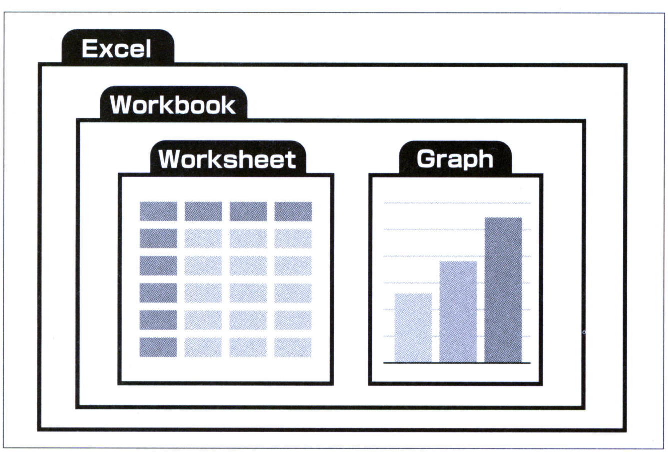
ちょっとややこしいのですが，Rangeオブジェクトを使用するためには，Rangeプロパティなどを使ってセルを特定する必要があります（第8図)。

Rangeプロパティを使って，Range("A1")としてRangeオブジェクトを使用します。

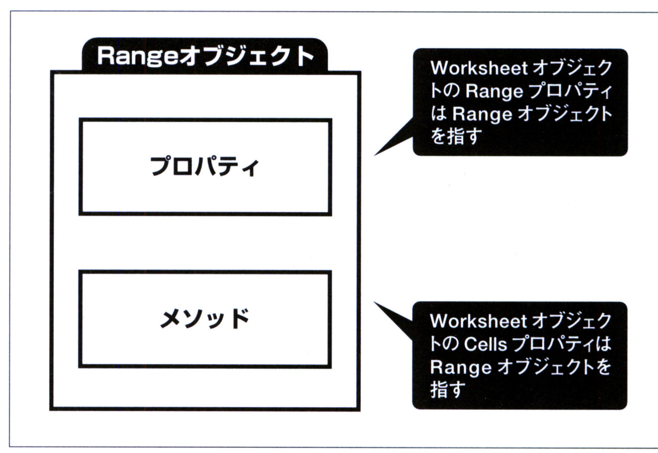
たとえば，「test.xlsのSheet1のセルA1の値」は

```
Workbooks（"test.xls"）.Work
sheets（"Sheet1"）.Range（"A1"）
.Value
```

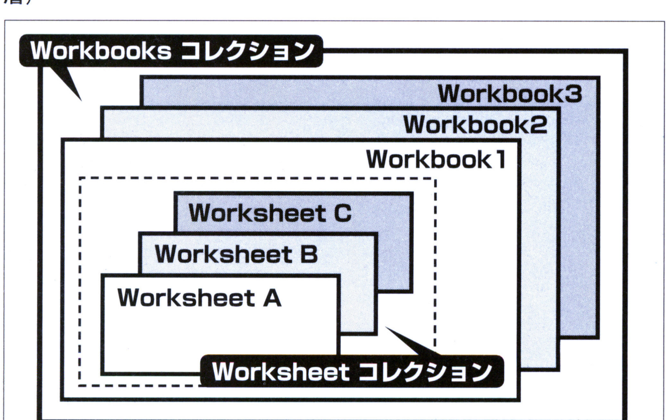
と指定することで取り出すことができます（第9図)。なお，使用するブックが，Excel VBAを記述しているブックと同



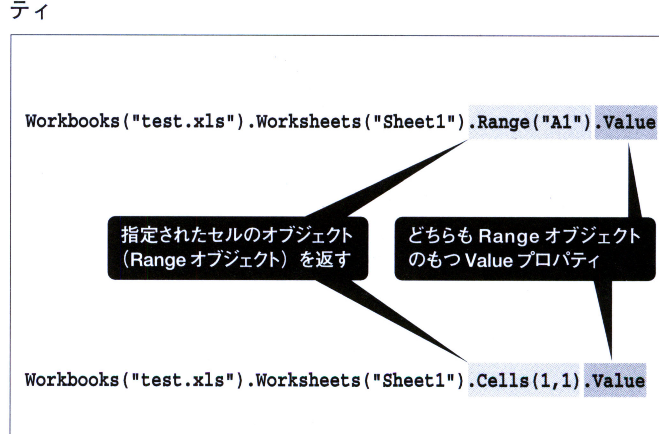
《第6図》オブジェクトの中に別のオブジェトがある場合もある（階層）



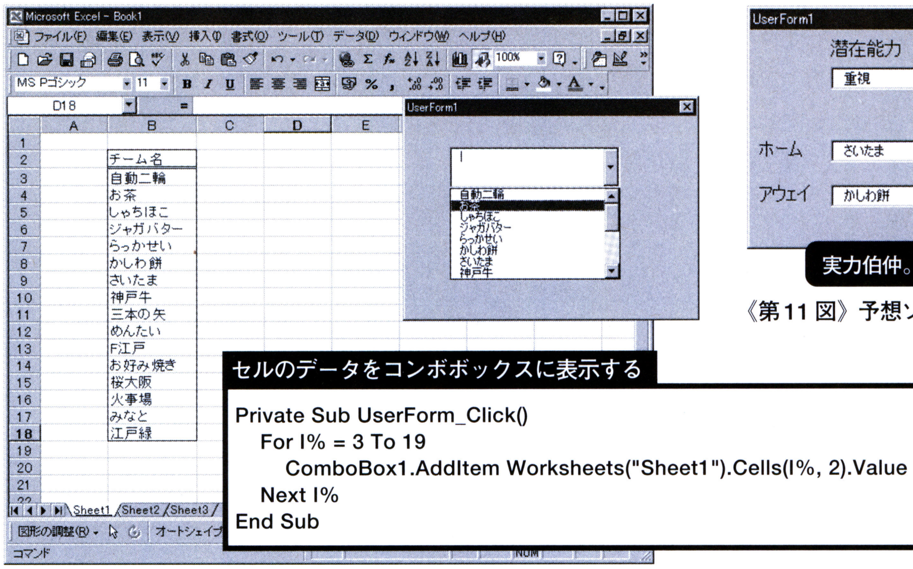
《第8図》Range オブジェクトと Range プロパティと Cells プロパティ



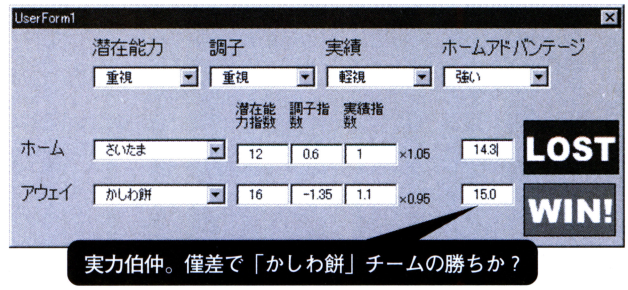
《第7図》同じ種類のオブジェクトをまとめ，扱いやすくしたのがコレクションだ



《第9図》Range オブジェクトの使用例



《第10図》オブジェクト（コレクション）とプロパティをうまく使うと簡単にセルの内容が取り出せる



《第11図》予想ソフトの完成予想図だ

じであれば，ブックの指定は省略できます。

Rangeプロパティは一つのセルや複数のセルを指定できる便利なプロパティです。しかし，セルの名前を文字列で指定するので，セルの値をつぎつぎに取り出したいときは不便です。そんなときは，Cellsプロパティを使用します。Cellsプロパティはセルの行と列のインデックスを指定します。

```
Workbooks ("test.xls") .Work
sheets ("Sheet1") .Cells (1,1)
.Value
```

とするとセル (1,1) に入力されたデータを取り出すことができます（第9図）。

コンボ ボックスにセルのデータを表示する

さて，対戦する2チーム名をすばやく選択，表示するにはコンボ ボックスが便利です。コンボ ボックスにデータを表示する方法は先月号で説明しましたので，おぼえている読者も多いはずですね!? コンボ ボックスにデータを追加するにはAdditem メソッドを利用します。

```
ComboBox1.AddItem " 日本 "
ComboBox1.AddItem " 韓国 "
```

セルのデータをコンボ ボックスに追加するためには，Additemの後にセルのデータを指定すればよさそうですね。したがって，

```
ComboBox1.AddItem Work
sheets("Sheet1").Range("A1")
.Value
```

と記述すればOKです。また，Cellsプロパティを使用して

```
ComboBox1.AddItem Work
sheets ("Sheet1") .Cells (1,1)
.Value
```

とすることもできます。

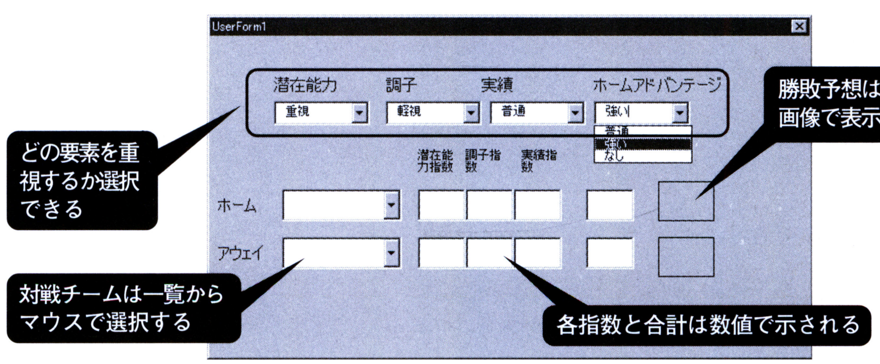
仮にセルB3～B19にチーム名が順に記載されている場合，それらのデータをすべてコンボ ボックスに追加するためには，つぎのようにプログラミングします（第10図）。

```
For I% = 3 to 19
    Combobox1.Additem Work
sheets ("Sheet1") .Cells (I%,2)
Next  I%
```

あとは仕上げるだけ

これで目標の予想プログラム（第11，12図）の下準備は整いました。あとは，各要素のポイントを計算／表示して，結果を画像で表示すると完成です。しかし，プログラミングする部分が多いので，次回に説明を加え完成させることにしましょう。

では，来月をお楽しみに。



《第12図》こんなイメージの予想プログラムをつくってみよう

Low Techinical Column

Excel VBAの投稿急増中!?

7月号の「4らいん」（原田淳史さん作）や8月号の「CUBEST」（NAKさん作）などExcel VBAの投稿が徐々に見られるようになりうれしい限りです。今後もいろいろな作品の投稿をお願いします。記事に関するリスエストや感想があれば，ぜひ愛読者カードやベーマガホームページ(http://www.basicmagazine.net/）の1行感想掲示板でお寄せください。


▲X君

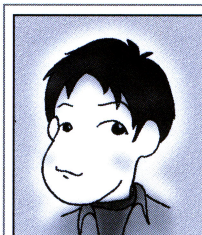
マイクロソフトの技術者がDirectXテクノロジーを伝授する！

IE+DirectAnimation マルチメディア・プログラミング

第6回　スプライト・アニメーションでものを動かす！

『Microsoft Internet Explorer（以下IEと略）』さえあれば，『DirectAnimation＊1』を使って誰でも簡単にマルチメディア・プログラムが作れます。そのテクニックを紹介する第6回はスプライト処理を取り上げて，写真1のようなサンプルを作成してみましょう。

TEXT&PROGRAMING●平井昌人



横浜市生まれの&H25歳男性。1993年4月に，開発言語製品のサポート・エンジニアとしてマイクロソフトに入社。現在はシステム・エンジニア（SE）としてVisual Studioを中心とした開発系の啓蒙・支援活動を行っている。先日，仕事でマイアミに10日ほど行ってきたが，終日カンヅメ状態でエメラルド・グリーンの海をながめただけに終わってしまった。

“スプライト”に飲み込まれるな！

今回は，2Dアニメーションの真髄ともいうべき，“スプライト”の処理を取り上げます。

スプライトといえば，あの清涼飲料水を思い浮かべる人もいるかもしれません（笑）。プログラミングの世界では，画面の中にある物体，敵や味方などキャラクターのことを意味します。通常，スプライトは動きをともない，いくつかのフレームを連続表示することで，キャラクターの複雑な動きや変化を表現しているのです。

とはいっても，むずかしいことはありませんので，入門者の人もぜひチャレンジしてください。今回は，**写真1**のように30枚のX君のJPEGファイルを連続表示して，アニメーションさせるサンプルを作ってみます。

スプライト・アニメーションの基本を押さえよう！

DirectAnimationには，イメージに“移動”“伸縮”“回転”させるなど，いろいろな動きをあたえるメソッドが多数用意されています。しかし，キャラクターが走ったり，ニッコリ笑ったり，また，敵機が爆発したりするなどの複雑な動きや変化を表現することは不可能です。

このように単純な数学的変換（移動／伸縮／回転）だけで表現しきれないことは，あらかじめ用意している画像を連続表示させることで実現するのです。まさしく“アニメーション”の仕組みと同じ考えかたで，これが本当の意味での2Dアニメーションといえます。

つまり，何枚もの静止画を切り替えて動いているように見せる原理は，なつかしのパラパラ漫画＊2と同じなのです。コマ送り撮影した何枚もの画像を準備しておき，それをDirectAnimationで高速に連続表示することで，なめらかで複雑な動きを演出することができます。

DirectAnimationでこのようなスプライト・アニメーションを実現する方法として，つぎの二つの代表的なパターンがあります。

①クリッピングを利用したアニメーション
②モンタージュを利用したアニメーション

それでは，一つずつ紹介していきましょう。

①クリッピングを利用したアニメーション

第1図のように，1枚のイメージの中に連続してキャラクターが並べられている画像ファイルを取り扱う場合は，“クリッピング”というテクニックを使います。クリッピングとは，【clipping：切り抜く】という意味です。つまり，イメージの中から1フレームぶんだけを切り抜き，表示するという手法です。

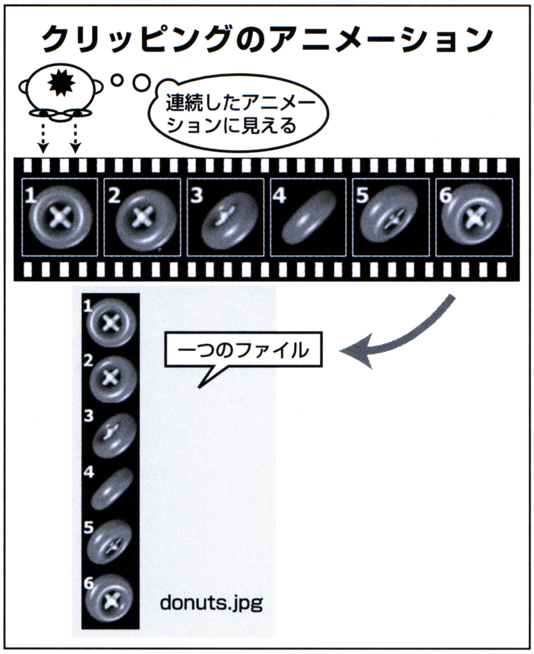
一般的なアニメーション・プログラミングでは，ビットマップから1フレームぶんをコピーして表示させるような手法を使います。しかしDirectAnimationでは，**第2図**のようにコピー



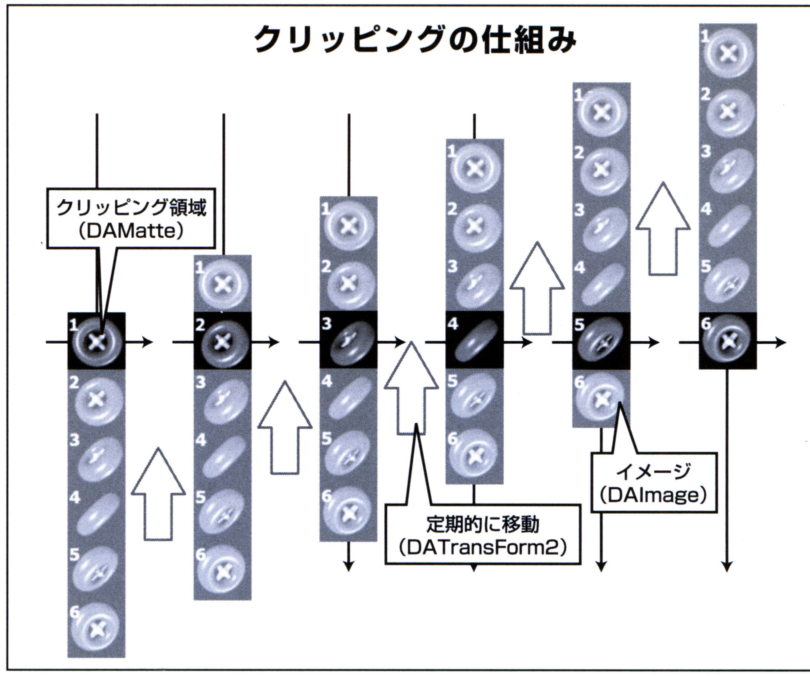
《**写真1**》今回作成するサンプル『スプライト・アニメーションX君』

＊1　DirectAnimation
2Dや3Dグラフィックス，サウンド，映像などを総合的にサポートする高レベルなAPIを提供している。IEに統合されており，インターネットでの利用を強く意識した設計となっている。

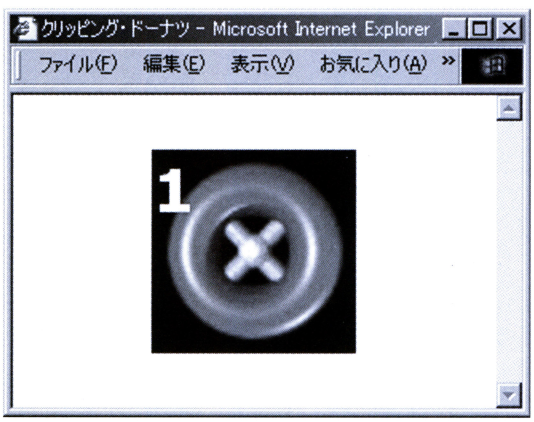
＊2　パラパラ漫画
教科書やノートの片隅に連続したキャラクターなどを書いて，ページをすばやくパラパラとめくると，キャラクターが動いて見えるといったもの。1秒間に30枚程度のスピード（これをフレーム・レートという）であれば，人間の目にはスムーズに動いて見える。画像の内容によっては，1秒間5～10フレームくらいでも違和感なくスムーズに見える場合もある。逆に，1秒間30フレーム以上にしても，人間の目がついてこないのでほとんど効果がない。



《第1図》“donuts.jpg”は，このように一つのファイルの中に連続したフレームを含んでいる



《第2図》クリッピングを利用したスプライト・アニメーションは，このように窓として表示している



《写真2》サンプル『クリッピング・ドーナツ』の実行画面

するのではなく，画像の1フレームのサイズを切り抜いておき，そこを窓として表示します。

それでは，サンプル・プログラム『クリッピング・ドーナツ』をベースに解説していきます。ソースコードはリスト1で，実行画面は写真2のようになります。

イメージファイルの読み込み～ImportImageメソッド

あらかじめ，ImportImageメソッドで画像ファイルを読み込んでおきます。通常，キャラクターは非矩形なので，背景を透明にする必要がありますね。透過色が設定されているGIFファイルであればImportImageメソッドでOKですが，その他の場合はImportImageColorKeyメソッドで透過色を指定します。

つぎのコードは，第1図の1フレーム128×128ピクセルで縦に6個のドーナツが並んだ“donuts.jpg”というJEPGファイルをロードします。なお，画像にはわかりやすいように，わざとフレーム番号を描き込んであります。

```
Set img = m.ImportImage("donuts.jpg")
```

フレーム・カウンタ用の変動値を作成

クリッピングは，第2図のように画像の中央で行うため，時間の経過とともに対応するフレームが中央にくるようイメージ自体を移動させる必要があります。

まずは，つぎのようにInterpolateメソッドで，0～6まで1.5秒で変位する変動値を作成します。これは，0からフレームの数まで増加，RepeatFor everメソッドで無限繰り返しの変動値となります。

```
Set num = m.Interpolate(0, 6, 1.5).RepeatForever
```

この変動値をもとに，おなじみのTranslate2Animメソッドで移動させます。X方向，Y方向に別々にfx, fyというDANumber型の変動値を計算します。

ここでは，多少複雑な計算を行っています。DirectAnimationでの演算は，第1表を参考にしてください。

ここの場合は，Floorメソッドにより小数部

《第1表》 演算に関するDirectAnimationのメソッド

VBScript	DirectAnimation	説明
a ＋ b	m.Add(numA, numB)	numAとnumBのDANumber型の加算
a － b	m.Sub(numA, numB)	numAとnumBのDANumber型の減算
a ＊ b	m.Mul (numA, numB)	numAとnumBのDANumber型の乗算
a ／ b	m.Div (numA, numB)	numAをnumBで割るDANumber型の除算
a Mod b	m.Mod(numA, numB)	numAをnumBで割った余りDANumber型の剰余
Int(a)	m.Floor(numA)	numAの小数部を切り捨てる

の切り取りを行っています。Floorメソッド*3を使わないと，**写真3**のようにルーレットみたいに画像のスクロールが見えてしまいます。

クリッピング領域を定義 ～FillMatteメソッド

DirectAnimationでは，切り取り（クリッピング）をするために，DAMatte*4というクリッピング領域を管理するオブジェクトが用意されています。

DAMatteオブジェクトは，DAPath2オブジェクトから定義します。サンプルでは，1フレームのサイズである128×128ピクセルの正方形でクリッピングします。

つぎのコードは，Rectメソッドで128×128の矩形の軌跡を定義して，FillMatteメソッドで軌跡（DAPath2オブジェクト）*5からクリッピング領域（DAMatteオブジェクト）を生成します。

```
Set matte = m.FillMatte(m.Rect(128, 128))
```

イメージにクリッピングを設定 ～Clipメソッド

クリッピングの領域（DAMatteオブジェクト）が定義できれば，実際のイメージ（DAImageオブジェクト）に適用すればいいわけです。クリッピング対象のDAImageオブジェクトのClipメソッドに，DAMatteオブジェクトを引き渡します。この処理によりイメージは，中央の128×128の部分だけしか見えなくなります。

```
Set img = img.Clip(matte)
```

②モンタージュを利用したアニメーション

さて，もう一つのスプライト・アニメーションの方法を紹介しておきましょう。

クリッピングによるアニメーションは，一つのファイルにすべてのキャラクターのフレームが含まれている場合に便利です。しかし，**第3**



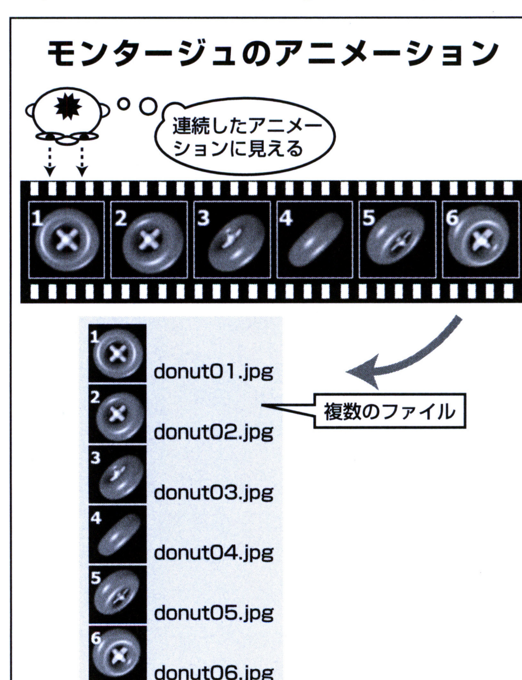
《写真3》 Floorメソッドにより整数倍で増減をさせないと，スクロールしてしまうのだ

図のように各フレームのキャラクターが一つ一つのファイルに分割されている場合は使えません。

このようなケースでは，“モンタージュ*6”というテクニックを使います。DirectAnimationではこのような用途のために，DAMontageというオブジェクトが用意されています。DAMontageオブジェクトは，複数の画像を格納しておき，それぞれの画像の上下関係により階層化することができます。

つまり，**第4図**のように上下関係を示すインデックスを変更することで一番上に表示する画像を動的に変更することができ，まさにスプライト・アニメーションの仕組みとなります。

それでは，サンプル・プログラム『モンタージュ・ドーナツ』をベースに解説していきます。ソースコードは**リスト2**で，実行結果は**写真2**の『クリッピング・ドーナツ』とまったく同じです。



《第3図》 “donut01.jpg” ～ “donut06.jpg” は，一つのファイルが1フレームで複数のファイルから構成される

＊3 Floorメソッド
小数点付きの数値に対し，小数点以下を切り捨てて整数に変換するメソッド。クリッピングに利用する変動値は，ストップ・ウォッチのように細かく刻むのではなく，「0,0,0,1,1,1,2,2,2,…」とする必要がある。つまり，時間とともに変化するけど，一定の期間同じ値が続くように変化させなければならないのだ。移動量に小数を残しておくと，ドーナツとドーナツの間への移動してしまう。時間の経過とともに「0.0→0.5→1.0」でなく，小数部を切り捨てた「0→0→1」のように変化する値が必要なのだ。

＊4 DAMatte
DAMatteをそのまま読むと「ディエー・マッテ」となり，ちょっと笑ってしまう。

＊5 軌跡（DAPath2オブジェクト）
図形や線，文字などの“軌跡”を管理するオブジェクト。軌跡とは，数学でいう一次変換で表現できる点や線の集合体。点や線を表す方程式のようなものを管理している。簡単にいえば，「DAPath2オブジェクトは図形や文字を描画するための型紙」だ。軌跡（DAPath2オブジェクト）からイメージ（DAImageオブジェクト）を描画するしくみは，型紙にそって鉛筆でなぞると，その図形が書けるのと同じ。

＊6 モンタージュ
みなさんは“モンタージュ写真”という言葉を聞いたことがあるかな？　犯人の人相を作るために，いろんな人の顔のパーツを組み合わせて作ったのが，モンタージュ写真。モンタージュ【montage：組み立て】という意味がある。

イメージ・ファイルの読み込み～ImportImageメソッド

モンタージュの場合，複数の画像ファイルを読み込む必要があります。このため，画像はループ・カウンタなどで簡単に表現できるファイル名を付けておきます。

サンプルでは“donut01.jpg”“donut02.jpg”…というように，連番を付けています。連続的に処理するため，Forループの中でつぎのようにカウンタ値から生成したファイル名を使ってロードします。

```
file = "donut" & Right("00" & n
+ 1, 2) & ".jpg"
Set img = m.ImportImage(file)
```

モンタージュの上下関係をきめる変動値の作成

モンタージュには，複数の画像が含まれ，その中での上下関係を示すインデックスの値が重要となります。インデックスの値が大きい画像ほど，前面になります。

スプライト・アニメーションでは，フレームの順番どおりに一番上（最前面）にそのフレームがくるような循環する変動値を生成しなければなりません。モンタージュは，この部分が非常にめんどうです。

サンプルでは，第5図のようにCosカーブ[*7]の性質を用いて，各フレームのインデックスが入れ替わり変動するように設定しています。

```
a = PI * 2 / 6 * (6 - n)
Set depth = m.Cos(m.Add(num,
m.DANumber(a)))
```

モンタージュの作成～ImageMontageAnimメソッド

つぎに，イメージ（DAImageオブジェクト）をもとにモンタージュ（DAMontageオブジェクト）を作成します。ImageMontageメソッドでは上下関係が固定されてしまうので，Animの付いたImageMontageAnimメソッドのほうを使います。

つぎのコードは，imgTmpイメージからモンタージュ（monTmp）を作成します。ImageMontageAnimメソッドの第2引数には，モンタージュ内での画像の上下関係を示すインデックスを指定します。サンプルでは，前述した循環する変動値（depth）をあたえています。

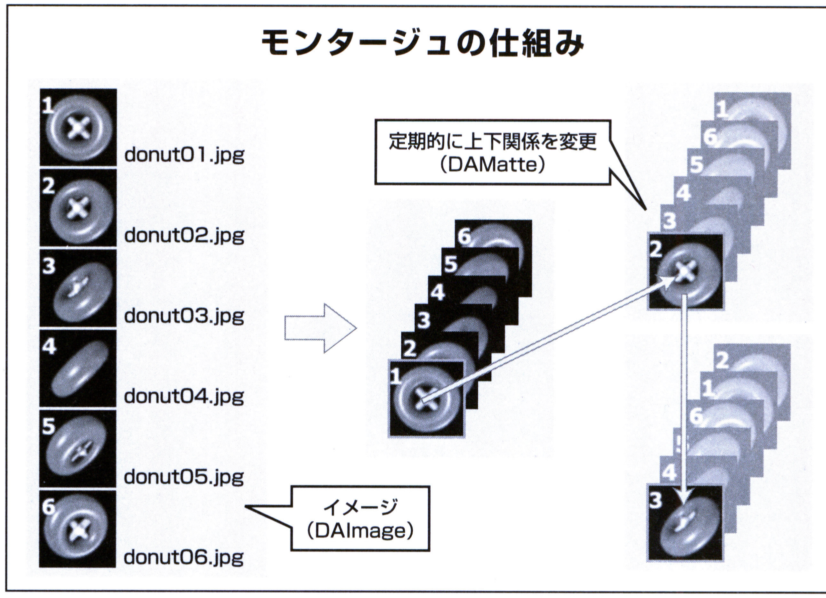
```
Set monTmp = m.ImageMontage
Anim(imgTmp, depth)
```

モンタージュを組み合わせる～UnionMontageメソッド

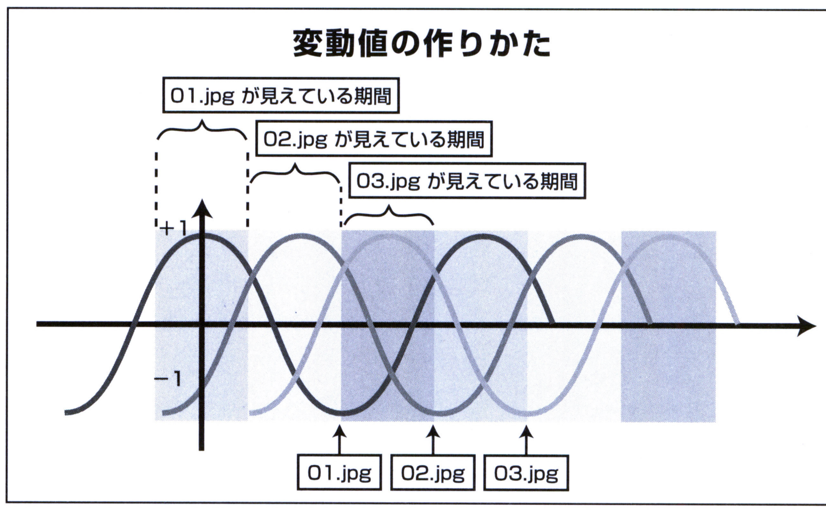
モンタージュは，イメージごとにその上下関係をインデックスで指定しながら作成します。これらのモンタージュを，UnionMontageメソッドで組み合わせていきます。Overlayメソッドとちがい，引数の順番は関係ありません。

つぎのコードは，既存のモンタージュに新しいモンタージュ（monTmp）を追加します。

```
Set mont = m.UnionMontage(monTmp, mont)
```

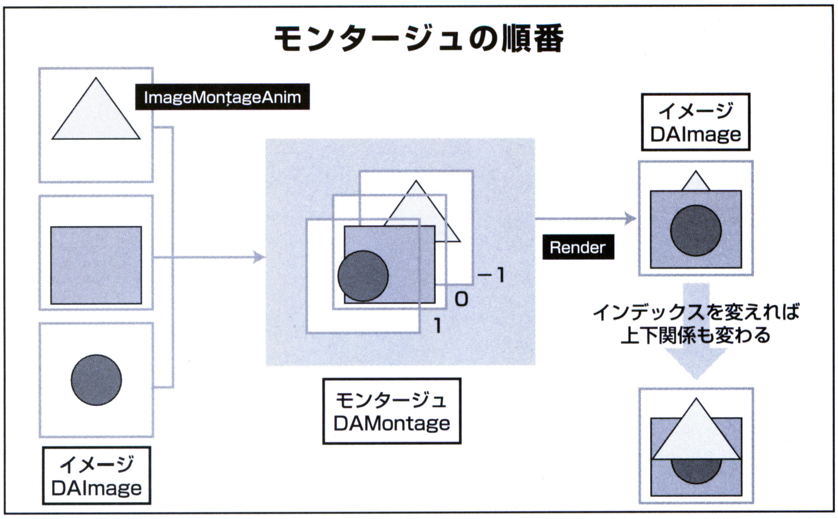


《第4図》モンタージュを利用する場合は，上下関係を示すインデックスを変更して行う

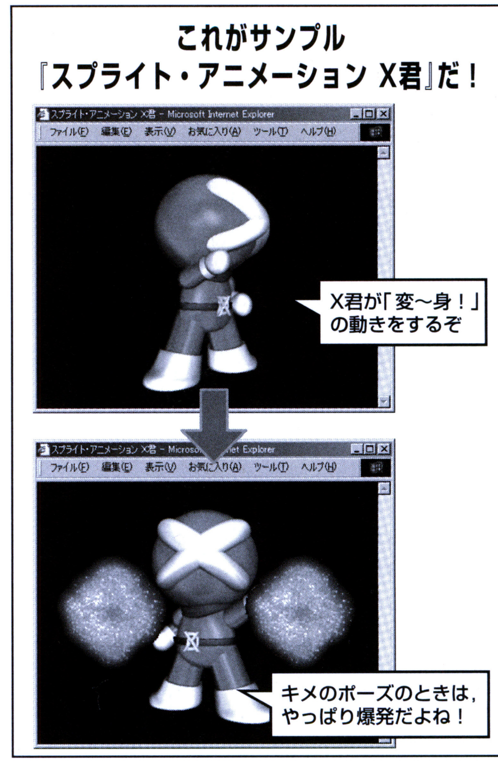


《第5図》このような仕組みで，モンタージュの上下関係を示すインデックス値を変位させている

＊7 Cosカーブ（余弦波）
横軸に角度（ラジアン）a，縦軸に「Cos（a）」を設定したグラフが，規則正しい波のような上下のカーブを描くためこのような呼びかたをする。Cos（a）は，「−1～1」までの範囲で変位する。しかし，均等に変位するわけでなく，0付近で高速に，−1と1付近で低速になる特徴がある。このため，いろいろな動きのもととなる変動値として利用するケースが多い。「Cos（0）＝1」なので，＋1からスタートする。これに対して「Sin（a）」による波形を，サインカーブ（正弦波）という。



《第6図》モンタージュは画像の合成に似ているが，上下関係が自由に変えられる



《第7図》X君がスプライト・アニメーションしたあと，爆発のスプライト・アニメーションと爆発音がする

モンタージュから再度イメージへ 〜Renderメソッド

何枚もの画像が組み込まれたモンタージュは，Renderメソッドにより再びイメージ（DAImageオブジェクト）へと変換されます。このときのイメージは，モンタージュの中でインデックスの順番で上下に並び替えられて，合成イメージとしてレンダリングされます。

たとえば，**第6図**のように三角，四角，円を組み合わせたモンタージュをレンダリングすると，その順番どおりのイメージが作成されます。このサンプルは，サイズが同じでかつ透過色を指定しないので，インデックスが大きいモンタージュのイメージだけが表示される結果となります。

つぎのコードは，モンタージュをレンダリングしてイメージを出力します。

```
Set img = mont.Render
```

サンプル・プログラム『スプライト・アニメーションX君』を見てみよう！

今回のテーマ，スプライト・アニメーションの処理を取り入れたサンプル・プログラム『スプライト・アニメーションX君』を用意しました。ソースコードは，**リスト3**です。

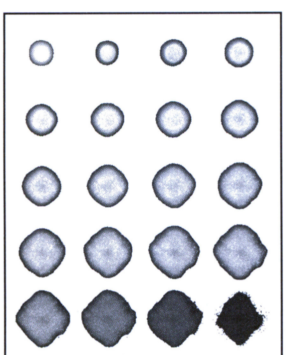
なお，サンプルのソースコードや画像ファイル，サウンド・ファイルなどは，ベーマガ・ホームページにありますのでダウンロードしてください。

●http://www.basicmagazine.net/

実行すると，**第7図**上のように，中央でポーズをきめるX君のアニメーションが再生されます。X君が最後にポーズをきめると，**第7図**下のように爆発音[*8]とともにX君の左右に爆発アニメーションが展開されます。

X君のアニメーションは，モンタージュを使い，爆発イメージはクリッピングの技法で処理しています。**第2表**も参照してください。

*8　爆発音
DirectAnimationでのサウンド処理に関しては，先月号でくわしく取り上げているので参考にしてほしい。音があるのとないのとでは，雰囲気がぜんぜんちがうからね。



《写真4》これが爆発イメージ用のGIFファイルだ



《写真5》X君イメージ用のJPEGファイル

まとめ

さて，今回は2Dアニメーションでのスプライト処理のテクニックを紹介しましたが，いかがでしたでしょうか。

アニメーションのもとになる画像をデザインするには，それなりのセンスと技術が要求されますが，連続画像さえあれば単純なコードでいとも簡単にアニメーションを作成できるのです。これまで紹介したDirectAnimationの機能と組み合わせれば，より複雑で高度なアニメーションに発展させることができます。

DirectAnimationのように「誰でも・簡単・高機能」と三拍子そろっているお気軽コンポーネントは，他にはありません。まだ一度も動かしてみたことのない人は，だまされたと思ってためしてみてください！

《第2表》サンプルで使用しているイメージ・ファイル

種類	ファイル名	内容
イメージ	bomb.gif	**写真4**のような爆発イメージ用のGIFファイル（透過色は設定済み）。1フレーム120×120で4列×5行＝計20フレーム
	X01.jpg～X30.jpg	**写真5**のようなX君イメージ用のJPEGファイル。300×300のサイズで，合計30ファイル
サウンド	bomb.wav	1.5秒の爆発音のウェーブ・ファイル

リスト1～リスト3入力＆実行方法

このリストは，IE5／5.5とテキスト・エディタ（『メモ帳』など）があれば，実行することが可能だ。IE5は，以下のアドレスから無償でダウンロードすることができる。

●**http://www.microsoft.com/windows/ie_intl/ja/download/**

リストをメモ帳などで入力したら，拡張子を"htm"または"html"にして保存する。そして，IE5の[ファイル]メニューより，[開く]で保存したファイルを読み込もう。すると，リストが実行されるようすを，IE5で確認することができる。

《リスト1》サンプル『クリッピング・ドーナツ』"cpDonut.htm"（VBScript用）

```
<HTML>
<TITLE>クリッピング・ドーナツ</TITLE>
<BODY LEFTMARGIN = "0", TOPMARGIN = "0">
<SCRIPT LANGUAGE = "VBScript">
'//--------------------------------------------------------------
'//
'// DirectAnimation Sample                        by Masato Hirai
'//
'//    『クリッピング・ドーナツ』
'//
'//--------------------------------------------------------------
Option Explicit

'//--------------------------------------------------------------
'// 初期化処理
'//--------------------------------------------------------------
Sub Window_onLoad()
  '//
  '// 変数・定数の宣言
  '//
  Dim m        'As DAStatics  'DAStaticsオブジェクト
  Dim img      'As DAImage    '最終イメージ
  Dim num      'As DANumber   '変動値
  Dim fx       'As DANumber   'X方向の移動量
  Dim fy       'As DANumber   'Y方向の移動量
  Dim matte    'As DAMatte    'クリッピング用のDAMatte
  '//
  '// DirectAnimationの初期化 (ピクセル座標系)
  '//
  Set m = DAControl.PixelLibrary
  '//
  '// イメージファイルの読み込み
  '//
  Set img = m.ImportImage("donuts.jpg")
  '//
  '// フレームカウンタ([1]→[6]フレームを1.5秒で)
  '//
  Set num = m.Interpolate(0, 6, 1.5).RepeatForever
  '//
  '// X方向の移動量(1列しかないので移動不要)
  '//
  Set fx = m.DANumber(0)
  '//
  '// Y方向の移動量([768]→[640]→[512]→,...の順)
  '//
  Set fy = m.Mul(m.DANumber(128), m.Floor(num))
  Set fy = m.Sub(m.DANumber(128 * 6 / 2 - 64), fy)
  '//
  '// イメージの移動(順番にフレームが中央にくるように)
  '//
  Set img = img.Transform(m.Translate2Anim(fx, fy))
  '//
  '// クリッピング領域の設定
  '//
  Set matte = m.FillMatte(m.Rect(128, 128))
  '//
  '// イメージにクリッピングを適用
  '//
  Set img = img.Clip(matte)
  '//
  '// コントロールにオブジェクトをセットしてスタート
  '//
  DAControl.Image = img
  DAControl.Start
  Call Window_onResize()
End Sub

'--------------------------------------------------------------
'ウィンドウのサイズ変更時の処理
'--------------------------------------------------------------
Sub Window_onResize()
  On Error Resume Next
  DAControl.Width  = Document.Body.ClientWidth
  DAControl.Height = Document.Body.ClientHeight
End Sub

</SCRIPT>
<! DirectAnimation コントロール>
<OBJECT ID = "DAControl"
  CLASSID = "CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D"
  STYLE = "position:absolute; left:0; top:0; z-index:-1">
</OBJECT>
</BODY>
</HTML>
```

《リスト2》サンプル『モンタージュ・ドーナツ』"moDonut.htm"（VBScript用）

```
<HTML>
<TITLE>モンタージュ・ドーナツ</TITLE>
<BODY LEFTMARGIN = "0", TOPMARGIN = "0">
<SCRIPT LANGUAGE = "VBScript">
'//--------------------------------------------------------------
'//
'// DirectAnimation Sample                        by Masato Hirai
'//
'//    『モンタージュ・ドーナツ』
'//
'//--------------------------------------------------------------
Option Explicit

'//--------------------------------------------------------------
'// 初期化処理
'//--------------------------------------------------------------
Sub Window_onLoad()
  '//
  '// 変数・定数の宣言
  '//
  Const PI = 3.14159265359  '円周率
  Dim m        'As DAStatics  'DAStaticsオブジェクト
  Dim img      'As DAImage    '最終イメージ
  Dim num      'As DANumber   '変動値
  Dim monTmp   'As DAMontage  '作業用モンタージュ
  Dim mont     'As DAMontage  'X君のモンタージュ
  Dim depth    'As DANumber   'モンタージュの切り替え制御
  Dim a        'As Double
  Dim file     'As String
  Dim n        'As Long
  '//
  '// DirectAnimationの初期化 (ピクセル座標系)
  '//
  Set m = DAControl.PixelLibrary
  '//
  '// フレームカウンタ用の変動値([0]→[360]を1.5秒で)
  '//
  Set num = m.Interpolate(0, PI * 2, 1.5).RepeatForever
  '//
  '// 空のモンタージュを定義
  '//
  Set mont = m.EmptyMontage
  For n = 0 To 5
    '//
    '// イメージファイルの読み込み
    '//
    file = "donut" & Right("00" & n + 1, 2) & ".jpg"
    Set img = m.ImportImage(file)
    '//
    '// フレームカウンタの調整([-1]→[+1]の変位)
    '//
    a = PI * 2 / 6 * (6 - n)
    Set depth = m.Cos(m.Add(num, m.DANumber(a)))
    '//
    '// モンタージュの作成
    '//
    Set monTmp = m.ImageMontageAnim(img, depth)
    '//
    '// モンタージュの合成
    '//
    Set mont = m.UnionMontage(monTmp, mont)
  Next
  '//
  '// モンタージュからイメージにレンダリング
  '//
  Set img = mont.Render
  '//
  '// コントロールにオブジェクトをセットしてスタート
  '//
  DAControl.Image = img
  DAControl.Start
  Call Window_onResize()
End Sub

'--------------------------------------------------------------
'ウィンドウのサイズ変更時の処理
'--------------------------------------------------------------
Sub Window_onResize()
  On Error Resume Next
  DAControl.Width  = Document.Body.ClientWidth
  DAControl.Height = Document.Body.ClientHeight
End Sub

</SCRIPT>
<! DirectAnimation コントロール>
<OBJECT ID = "DAControl"
  CLASSID = "CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D"
  STYLE = "position:absolute; left:0; top:0; z-index:-1">
</OBJECT>
</BODY>
</HTML>
```

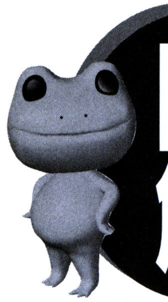
《リスト3》サンプル『スプライト・アニメーションX君』"X-Kun.htm"（VBScript用）

```
<HTML>
<TITLE>スプライト・アニメーション X君</TITLE>
<BODY LEFTMARGIN = "0", TOPMARGIN = "0">
<SCRIPT LANGUAGE = "VBScript">
'//--------------------------------------------------------------
'//
'// DirectAnimation Sample                        by Masato Hirai
'//
'//    『スプライト・アニメーション X君』
'//
'//--------------------------------------------------------------
Option Explicit

'//--------------------------------------------------------------
'// 初期化処理
'//--------------------------------------------------------------
Sub Window_onLoad()
  '//
  '// 変数・定数の宣言
  '//
  Const PI = 3.14159265359  '円周率
  Const COLS = 4            'イメージに含まれるキャラクタの列数
  Const ROWS = 5            'イメージに含まれるキャラクタの行数
  Const FRAMES = 20         'イメージに含まれるキャラクタの総数
  Const CW = 120            '1キャラクタの幅(ピクセル)
  Const CH = 120            '1キャラクタの高さ(ピクセル)
  Const FILES = 30          'イメージファイルの総数
  Const PLAY_TIME = 6       'アニメーションの1サイクル(秒)
  Const BOMB_DELAY = 3.5    '爆発までの遅延時間(秒)
  Dim m        'As DAStatics  'DAStaticsオブジェクト
  Dim num      'As DANumber   '変動値
  Dim fx       'As DANumber   'X方向の移動量
  Dim fy       'As DANumber   'Y方向の移動量
  Dim matte    'As DAMatte    'クリッピング用のDAMatte
  Dim monTmp   'As DAMontage  '作業用モンタージュ
  Dim mont     'As DAMontage  'X君のモンタージュ
  Dim depth    'As DANumber   'モンタージュの切り替え制御
  Dim imgTmp   'As DAImage    '作業用イメージ
  Dim xkun     'As DAImage    'X君イメージ
  Dim bombR    'As DAImage    '爆発イメージ(右)
  Dim bombL    'As DAImage    '爆発イメージ(左)
  Dim back     'As DAImage    '背景イメージ
  Dim img      'As DAImage    '最終イメージ
  Dim eff      'As DASound    '効果音(爆発音)
  Dim snd      'As DASound    '最終サウンド
  Dim a        'As Double
  Dim file     'As String
  Dim n        'As Long
  '//
  '// DirectAnimationの初期化 (ピクセル座標系)
  '//
  Set m = DAControl.PixelLibrary
  '//
  '// X君アニメーションの作成(モンタージュ)
  '//
  Set num = m.Interpolate(0, PI * 2, PLAY_TIME).RepeatForever
  Set mont = m.EmptyMontage
  For n = 0 To FILES - 1
    file = "X" & Right("00" & n + 1, 2) & ".jpg"
    Set imgTmp = m.ImportImage(file)
    a = PI * 2 / FILES * (FILES - n)
    Set depth = m.Cos(m.Add(num, m.DANumber(a)))
    Set monTmp = m.ImageMontageAnim(imgTmp, depth)
    Set mont = m.UnionMontage(monTmp, mont)
  Next
  Set xkun = mont.Render      '← モンタージュからイメージへ
  '//
  '// 爆発アニメーションの作成(クリッピング)
  '//
  Set imgTmp = m.ImportImage("bomb.gif")
  Set num = m.Sequence(m.DANumber(-1).Duration(BOMB_DELAY), _
           m.Interpolate(0, FRAMES, PLAY_TIME - BOMB_DELAY))
  Set num = num.RepeatForever
  Set fx = m.Mul(m.DANumber(CW), _
               m.Floor(m.Mod(num, m.DANumber(COLS))))
  Set fx = m.Sub(m.DANumber(CW * COLS / 2 - CW / 2), fx)
  Set fy = m.Mul(m.DANumber(CH), _
               m.Floor(m.Div(num, m.DANumber(COLS))))
  Set fy = m.Sub(m.DANumber(CH * ROWS / 2 - CH / 2), fy)
  Set imgTmp = imgTmp.Transform(m.Translate2Anim(fx, fy))
  Set matte = m.FillMatte(m.Rect(CW, CH))
  Set imgTmp = imgTmp.Clip(matte)
  '//
  '// 爆発アニメーションイメージを1.5倍に拡大
  '//
  Set imgTmp = imgTmp.Transform(m.Scale2Uniform(1.5))
  '//
  '// 爆発アニメーションイメージを左右に位置を変えて複製
  '//
  Set bombR = imgTmp.Transform(m.Translate2(120, 0))
  Set bombL = imgTmp.Transform(m.Translate2(-120, 0))
  '//
  '// 背景イメージ(黒のべた塗り)
  '//
  Set back = m.SolidColorImage(m.Black)
  '//
  '// イメージを合成して最終イメージを生成
  '//
  Set img = m.OverlayArray(Array(bombL, bombR, xkun, back))
  '//
  '// サウンド(効果音)の処理
  '//
  Set eff = m.ImportSound("bomb.wav").Sound
  Set snd = m.Sequence(m.Silence.Duration(BOMB_DELAY), _
                       eff.Duration(PLAY_TIME - BOMB_DELAY))
  Set snd = snd.RepeatForever
  '//
  '// コントロールにオブジェクトをセットしてスタート
  '//
  DAControl.Image = img                 '← イメージ
  DAControl.Sound = snd                 '← サウンド
  DAControl.Start
  Call Window_onResize()
End Sub

'--------------------------------------------------------------
'ウィンドウのサイズ変更時の処理
'--------------------------------------------------------------
Sub Window_onResize()
  On Error Resume Next
  DAControl.Width  = Document.Body.ClientWidth
  DAControl.Height = Document.Body.ClientHeight
End Sub

</SCRIPT>
<! DirectAnimation コントロール>
<OBJECT ID = "DAControl"
  CLASSID = "CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D"
  STYLE = "position:absolute; left:0; top:0; z-index:-1">
</OBJECT>
</BODY>
</HTML>
```



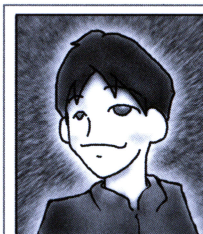
プログラマーをめざす人の必見講座

図解！ Visual Basic ゲーム作成術

第6回　シューティング・ゲームを作る③敵キャラクターに弾を撃たせる！

TEXT&PROGRAMMING●東　幸太

シューティング・ゲームでは，プレイヤー・キャラクターだけでなく，敵キャラクターも弾を撃ってくるのが普通です。敵キャラクターに体当たり攻撃以外の攻撃方法を持たせることで，敵キャラクターは無理にプレイヤー・キャラクターめがけて体当たりする必要がなくなり，多彩な移動アルゴリズムを組み込めます。今回は，敵キャラクターに弾を撃たせるテクニックを紹介し，『Microsoft Visual Basic（以下VBと略）』でサンプルを作成してみます。



BASIC歴15年以上のプログラマー。メモリの価格が安くなり，手軽に256Mバイトのメモリが買えるようになった。Windows NTや2000の場合は，メモリを増やせば増やしただけ快適になっていく。384M＋Windows 2000の環境に慣れると，もうWindows95／98／Meにはもどれないよ（笑）。

Microsoft Visual Basic Programming System for Windows Ver.6.0（日本語版）

オープン価格
※編集部の価格調査によると，下記価格が目安になります。
ラーニング版2万円強
（アカデミック・パッケージ1万円弱）
プロフェッショナル版8～9万円
（アカデミック・パッケージ1万円強）
問い合わせ先
マイクロソフト
☎03-5454-2300
ホームページ
http://www.microsoft.com/japan/
動作環境
Windows 95／98／2000／NT4.0が動作するマシン



敵弾が登場するとどんな効果がある？

たいていのシューティング・ゲームでは，プレイヤー・キャラクター（自機）だけでなく，敵キャラクターも弾を撃ってきますね。

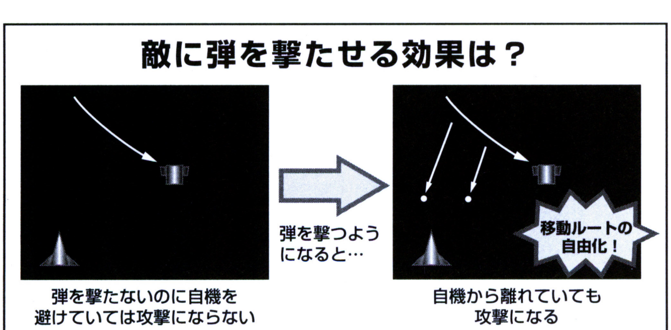
敵弾を追加することで，ただ単純に自機へ体当たり攻撃を仕掛けるだけでなく，弾によって遠方から自機を狙うという攻撃も可能となります。これによって，敵キャラクターの攻撃バリエーションが増加するわけですが，体当たりをしなくても済むようになることから，敵キャラクターの移動経路も自由になります。つまり，移動バリエーションも同時に増加するのです（第1図）。

敵弾には，大きくわけて「プレイヤーの弾によって相殺できるタイプ」と「相殺できないタイプ」があります（第2図）。通常は，弾同士がすれちがってしまって相殺できないタイプが利用されます。

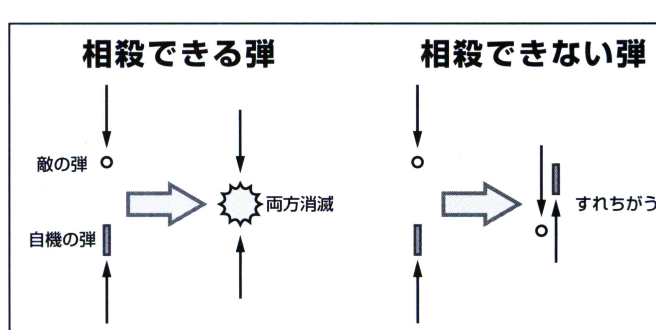
弾の種類が豊富な場合は，相殺できる弾も登場させる場合がありますが，通常は相殺できないようにしておきます。これは，プレイヤーの弾が敵弾によって相殺されてしまい，プレイヤーが正確なタイミングで弾を撃っても敵キャラクターに命中しないことによるストレスを防ぐためです。

この他，敵キャラクターと同様に，敵弾の移動アルゴリズムの調整でゲームの難易度を調整できます。自機の存在しない方向へ撃たせておけば，ゲームの雰囲気が変わる程度で難易度には影響しませんし，正確に狙わせれば難易度は一気に高まります。

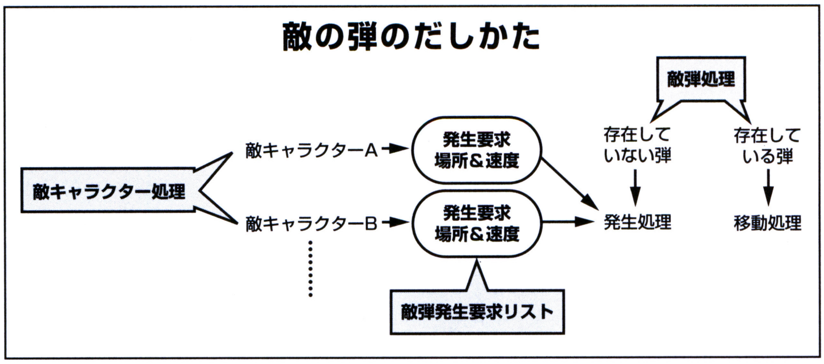
さらに，自機の移動する方向を予想して狙わせると，さらにむずかしいゲームになります。常時このような弾を登場させると，むずかしくなりすぎてゲームになりませんが，ときどき登場させるようにするとゲームの流れが単調にならず，



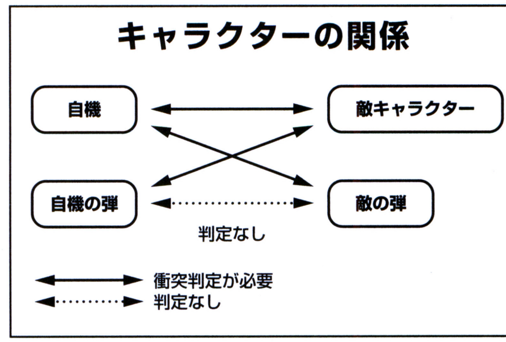
《第1図》このような敵の移動ルートが自由になること以外にも，ゲーム性を上げたり難易度を調整したりと，いろいろな効果が期待できる



《第2図》一般的なゲームでは弾を相殺できないが，特殊な弾のみ相殺できるようにしてもおもしろい効果がある



《第3図》今回作成するゲームは，敵弾発生要求リストによる発生処理を行っている



《第4図》衝突判定の処理には気をつけよう。弾を相殺できる場合は，すべての判定を行う必要がでてくる

おもしろみが増します。

敵弾はどうやって処理する？

敵弾は，同じ特性を持つキャラクターが複数個登場するタイプといえるでしょう。

そこで，個々の弾のデータは配列変数に格納しておき，処理プログラムは共通化してやります。処理の内容は基本的には敵キャラクターと同じで，個々の弾について順番に移動処理を行っていきます。

敵弾が敵キャラクターと異なる点の一つとして，発生するタイミングが敵弾側ではきまらないということがあります。敵弾は敵キャラクターから発射されるということで，敵キャラクター側で発射タイミングをきめるのが普通です。

弾のほうでタイミングをきめ，座標だけ敵キャラクターのものを参照する，ということも可能ですが，敵キャラクターのふるまいを考えると，敵キャラクターの位置などに応じて弾の発射タイミングをきめるほうが有利です。

このためには，敵弾の発生処理を敵キャラクター処理の部分で実行しなければなりません。この処理には，さまざまな方法が考えられます。ここでは，敵キャラクター処理プログラムから，発生要求リストを敵弾処理プログラムに渡し，具体的な発生処理自体は敵弾処理プログラム中で行うことにしましょう。

弾の「発生要求リスト」を活用する！

まず，敵キャラクター処理の部分で敵弾を発生させる場合，直接敵弾のデータを操作するのではなく，敵弾発生要求リストに発生させる敵弾のデータを追加していくようにします。

敵弾発生要求リストには，発生させる弾の座標と速度を記録しておきます。つぎに，敵弾処理の部分では，存在していない敵弾処理の部分で敵弾発生要求リストを順番にチェックしていき，敵弾をリストのデータに応じて初期化することで発生させます（第3図）。

これによって，敵キャラクター処理からだされる敵弾要求に対し，敵弾処理が応えて敵弾が発生するようになります。

この方式では，敵弾発生要求リストという配列変数が余分に必要となるものの，敵弾のデータ格納配列変数の“空き”を探す処理が省略できることから，処理速度の高速化が可能となります。記憶領域の制約がゆるく，実行速度にやや難点のあるVBの場合はこの方式が適しているといえるでしょう。

敵弾が追加されたシューティング・ゲームは，第4図のようなキャラクター関係になります。敵弾とプレイヤーの弾が相殺しないことにすると，敵弾と自機の間の衝突判定だけが増加することになります。

衝突判定自体は，キャラクターの大きさが異なるだけで，敵キャラクターとの衝突判定と同様です。

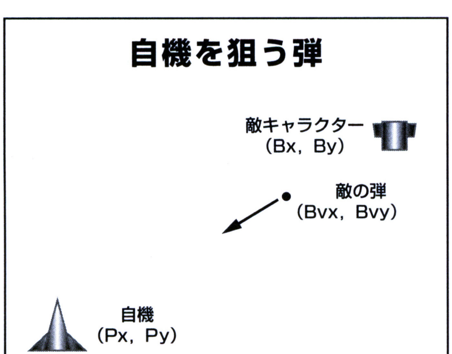
自機を狙う処理を考えてみよう

敵弾の移動パターンとしては，まっすぐ降下してくるものや，やや左右に変化するもの，自機の位置を正確に狙ってくるものなどがあります。この中で，自機を狙わせる場合には，速度の設定に工夫が必要です。

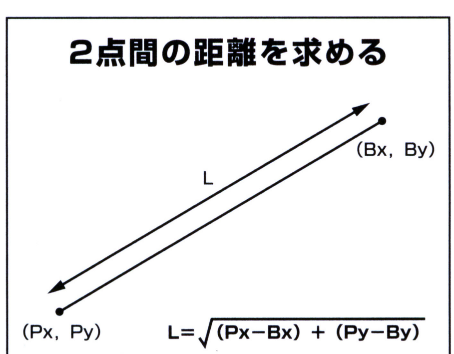
まっすぐ降下するものは，X方向の速度を0に設定すれば済みます。やや左右に変化させる場合は，X方向の速度を調整すればよいですね。しかし，自機の存在する座標に到達させる場合には，X方向の速度とY方向の速度をうまく設定する必要があります。

それでは第5図のように，自機の座標がPx，Py，敵弾の初期座標がBx，Byである場合に，敵弾が自機の座標に到達するような速度Bvx，Bvyを計算することを考えてみましょう。

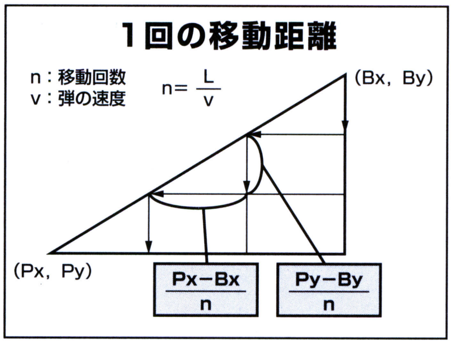
まず，最初に敵弾が自機の座標に到達するまでに何回移動処理を実行するかを計算します。第6図のように距離の計算を行えば，1回の移動距離で割ることで到達するまでに必要な移動回数がわかり



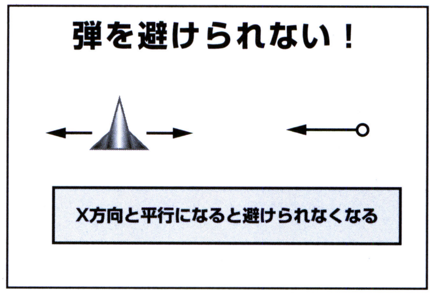
《第5図》自機を狙うために，速度計算を行う



《第6図》2点間の距離は，このような式で求める



《第7図》各方向の速度は，このように計算できる



《第8図》敵の弾がX方向と平行になるほど，自機はこれを避けにくくなる。自機が左右にしか動けない場合は，注意しよう

《第1表》『シューティング・ゲーム3』の主な変数

変数名	内容
hiscore	ハイスコア
score	プレイヤー・スコア
px	自機X座標
py	自機Y座標
ef()	敵キャラクター状態フラグ
ex()	敵キャラクターX座標
ey()	敵キャラクターY座標
evx()	敵キャラクターのX方向速度
evy()	敵キャラクターのY方向速度（降下速度）
et()	敵キャラクタータイプ（0，1，2の3とおり）
ei()	敵キャラクター敵弾発生間隔
eic()	敵キャラクター敵弾発生間隔カウンタ
ebf()	敵弾の状態フラグ
ebx()	敵弾のX座標
eby()	敵弾のY座標
ebvx()	敵弾のX方向速度
ebvy()	敵弾のY方向速度（降下速度）
ebix()	敵弾発射要求リスト：X座標
ebiy()	敵弾発射要求リスト：Y座標
ebivx()	敵弾発射要求リスト：X方向速度
ebivy()	敵弾発射要求リスト：Y方向速度
pbf()	プレイヤーの弾の状態フラグ
pbx()	プレイヤーの弾のX座標
pby()	プレイヤーの弾のY座標
firereq	弾の発生要求フラグ
trig0flag	発射キーを押しているかどうかのフラグ
ti	残り時間
gamestat	ゲーム進行状況（0：ゲーム中，1：ゲーム・オーバー）

ますね。X方向，Y方向それぞれの方向について，この回数だけ移動処理を行った結果，自機の座標に到達すればよいことになります。

よって，それぞれの方向について移動させる距離を計算し，移動回数で割ることでそれぞれの方向についての1回の移動距離＝速度を計算できます（**第7図**）。

このようにして速度を設定すれば，敵弾は発射された瞬間の自機の座標を正確に狙って移動するようになります。

ただし，自機が左右にしか移動できない場合は，自機が敵弾から逃げ切れない場合がでてくるので（**第8図**），敵キャラクターに敵弾を発射させる位置を制限するなどの工夫が必要です。

シューティング・ゲームで処理を確認してみよう！

それでは，実際にVBでプログラミングされたシューティング・ゲームを実行してみましょう。プログラムは，**リスト1**です。

リスト1を入力して実行すると，**第9図**のような画面が表示されます。画面には，プレイヤーが操作する自機が登場します。また，画面上部からの敵キャラクターが降下してきます。

プレイヤーは，自機をテンキーの[4][6]キーで左右に移動させ，敵キャラクターの体当たり攻撃と弾による攻撃を避けてください。そして，スペース・キーで弾を発射して，敵キャラクターを撃破しなくてはなりません。弾が敵キャラクターに命中すれば，敵キャラクターは撃破されて得点がアップします。

敵キャラクターや敵弾が自機に接触すると自機はダメージを受け，残りプレイ時間が減少します。また，撃破した敵キャラクターに接触した場合は，ダメージを受けません。

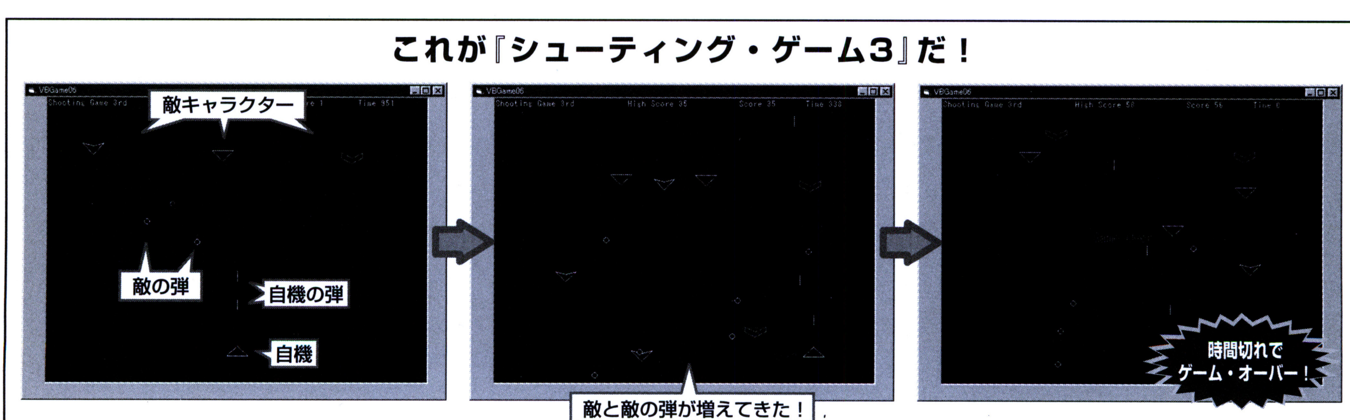
プレイ時間がなくなると，ゲーム・オーバーとなります。[Esc]キーで，ゲーム終了となります。

リスト中ではどう処理している？

このゲームでは，敵キャラクターが3種類登場します。そのうちの2種類が弾を発射しますが，一方の敵キャラクターの弾はまっすぐ降下し，もう一方の弾は自機を正確に狙ってきます。

この2種類の弾は，移動処理自体は完全に共通で，発射させるときの初期速度設定の方法を個別に設定することで，弾の移動パターンを変えています。X方向の速度を0にしてやればまっすぐ降下するだけになりますし，**第7図**のような計算を行えば自機の方向へ向けて移動するようになります。

敵キャラクター処理の中で，弾を発射するタイミングであると判断した場合は，敵弾発生要求リスト（ebix，ebiy，ebivx，ebivyの四つの配列変数）に敵弾データ



《第9図》発射される敵弾は，敵キャラクター・パターンによって異なり，真下に落下してくるものと自機を狙ってくるものの2種類がある。それぞれのパターンの特性を把握して，敵弾のコースを先読みして回避すると，高得点に結びつくぞ

を追加します。

リストには，各座標方向についての初期位置と速度が格納されます。敵弾処理の中で，存在しない敵弾が見つかったときには，このリストを参照して対応する変数(ebx，eby，ebvx，ebvyの四つの配列変数)を初期化してやります。

敵弾の個別情報については，敵キャラクター処理中で計算し，“空いている”敵弾用配列変数を探す部分は敵弾処理の部分で行う形になります。

まとめ

敵キャラクターに敵弾を発射させることで，ゲーム画面がにぎやかになり，敵キャラクターの移動経路も自由に設定できるようになります。

敵弾はゲームの難易度に大きく影響することから，敵弾の速度や数，移動方法などをうまく調整してやらないと，むずかしくなりすぎる場合があります。この点に注意しながら，いろいろためしてみてください。

《リスト1》『シューティング・ゲーム3』のプログラム・リスト(VB6.0用)

```
Option Explicit

Dim keystat%(9) 'テンキーの状態を記録する変数
Dim keytrig0%, keytrig1%, keytrig2%, keytrig3% 'トリガー用
キーの状態を記録する変数
Dim px!, py! '自機の座標
Dim  ex!(9),  ey!(9),  ef%(9),  et%(9),  evx!(9),  evy!(9),
ei%(9), eic%(9) '敵キャラクタ
Dim pbx!(2), pby!(2), pbf!(2) 'プレイヤーの弾
Dim ebx!(19), eby!(19), ebf%(19), ebvx!(19), ebvy!(19) '敵
弾
Dim ebix!(19), ebiy!(19), ebivx!(19), ebivy!(19) '敵弾発生
要求リスト
Dim hiscore%, score% 'ハイスコアとスコア
Dim ti% '残り時間
Dim damage% 'ダメージ回数
Dim gamestat% 'ゲーム進行状況
Dim trig0flag% 'トリガキー状態
Dim firereq% 'プレイヤーの弾の発射要求フラグ
Dim col&(2) '敵キャラクタのカラー
Dim ebin% '敵弾発生要求リスト項目数

Private Sub Form_Load()
    hiscore = 0
    gameinit
End Sub

Private Sub gameinit() 'ゲーム初期化
    Dim ox%, oy%, x%, y%, i%
    With Form1
        .Width = Screen.TwipsPerPixelX * 640 'フォームのサ
イズを設定
        .Height = Screen.TwipsPerPixelY * 480
        .Caption = "VBGame06"
    End With
    With Picture1
        .Width = Screen.TwipsPerPixelX * 576 'ピクチャボッ
クスのサイズを設定
        .Left = Screen.TwipsPerPixelY * 32
        .Height = Screen.TwipsPerPixelY * 432
        .Top = 0
        .ScaleMode = 3
        .ScaleHeight = 432
        .ScaleWidth = 576
        .BackColor = RGB(0, 0, 0)
        .ForeColor = RGB(255, 255, 255)
        .FillStyle = 0
        .Font = "ＭＳ ゴシック"
        .FontSize = 10
    End With
    Picture1.Cls
    Form1.Show

    col(0) = RGB(0, 255, 255)
    col(1) = RGB(255, 0, 255)
    col(2) = RGB(255, 255, 0)

    Picture1.Print "Shooting Game 3rd"

    px = 576 / 2 '自機の初期座標
    py = 416 - 32

    For i = 0 To 9
        ef(i) = 0
    Next

    For i = 0 To 19
        ebf(i) = 0
    Next

    For i = 0 To 2
        pbf(i) = 0
    Next

    score = 0
    ti = 1000
    putscore
    trig0flag = 0

    gamestat = 0 'ゲームモード（0でゲーム中、1でゲームオ
ーバー状態）

    Timer1.Interval = 30 'インターバルタイマ起動
End Sub

Private Sub Picture1_MouseUp(button%, shift%, x!, y!)
    If (button And 2) = 2 Then 'マウス右ボタンクリックで強
制終了
        End
    End If
End Sub

Private Sub Picture1_KeyDown(key%, shift%)
    If key = 27 Then 'ESCキーで強制終了
        End
    End If
    If key > 96 And key < 106 Then 'テンキー設定
        keystat(key - 96) = 1
    End If
    If key = 32 Then keytrig0 = 1 'トリガキー設定
    If key = 90 Then keytrig1 = 1
    If key = 88 Then keytrig2 = 1
    If key = 13 Then keytrig3 = 1
End Sub

Private Sub Picture1_KeyUp(key%, shift%)
    If key > 96 And key < 106 Then
       keystat(key - 96) = 0
    End If
    If key = 32 Then keytrig0 = 0
    If key = 90 Then keytrig1 = 0
    If key = 88 Then keytrig2 = 0
    If key = 13 Then keytrig3 = 0
End Sub

Private Sub Timer1_Timer()
    Dim npx!, i%, j%
    Dim npby!
    Dim nex!, ney!, dx!
    Dim x!, xerr!, v!, l!, d!
    Dim nebx!, neby!
    If gamestat = 0 Then '通常ゲームモード
        '自機の処理
        npx = px
        If keystat(4) = 1 Then 'キー状況に応じて新規座標を
変更
            npx = px - 5
        End If
        If keystat(6) = 1 Then
            npx = px + 5
        End If
        If npx < 20 Then npx = 20 '移動範囲制限
        If npx > 556 Then npx = 556
        putplayer px, py, 0 '元の場所のキャラクタを消去
         putplayer npx, py, RGB(255, 255, 255) '新しい場所
にキャラクタを描画
        px = npx '自機の座標を更新
        If keytrig0 = 1 Then '発射キーを押しているか？
             If trig0flag = 0 Then '一旦発射キーを離してい
るか？
                firereq = 1 'プレイヤーの弾の発生要求
                trig0flag = 1
            End If
        Else
            trig0flag = 0
        End If
        'プレイヤーの弾の処理
        For i = 0 To 2
            If pbf(i) = 0 Then
                 '存在していない場合は発射するかどうかを判
断
                If firereq = 1 Then
                    pbf(i) = 1 'プレイヤーの弾の発生
                    pbx(i) = px '自機の位置に発生
                    pby(i) = py
                    firereq = 0
                End If
            Else
                '存在している場合の処理
                npby = pby(i) - 13 '上昇
                If npby < 16 Then
                    pbf(i) = 0 '画面上まで来たら消滅
                    putpb pbx(i), pby(i), 0
                Else
                    For j = 0 To 9 '敵キャラクタとの衝突判
定
                        If ef(j) = 1 And ex(j) + 16 >
pbx(i) And ex(j) - 16 < pbx(i) And ey(j) + 16 > npby And
ey(j) - 16 < npby Then
                            ef(j) = 2
                            scoreup 1
                            pbf(i) = 0
                        End If
                    Next
                     putpb pbx(i), pby(i), 0 '元の位置から
消去
                    If pbf(i) = 1 Then '存在している時だけ
新しい場所に表示
                        putpb pbx(i), npby, RGB(255, 255,
0)
                    End If
                    pby(i) = npby '座標更新
                End If
            End If
        Next
        '敵キャラクタの処理
        ebin = 0 '敵弾発生要求リストクリア
        For i = 0 To 9
            If ef(i) = 0 And score / 3 + 3 > i Then
                '敵キャラクタ発生処理
                ef(i) = 1
                ex(i) = Rnd * 500 + 38 '乱数でX座標を決定
                ey(i) = 32 '画面上部から登場させる
                 et(i) = Int(Rnd * 3)    '乱数で敵キャラク
タのパターンを決定
                 ei(i) = 30 - score / 20 'スコアが増えるに
つれて敵弾発射間隔を短くする
                eic(i) = ei(i)
                evy(i) = Rnd * 3 + 3 '下降速度は乱数で調整
                If et(i) = 0 Then 'Type A
                     'Type Aの場合は最初のX方向の速度は自
機を避ける方向
                    evx(i) = Rnd + 2
                    If px > ex(i) Then evx(i) = -evx(i)
                ElseIf et(i) = 1 Then 'Type B
                     'Type Bの場合は最初のX方向の速度は乱
数
                    evx(i) = Rnd * 5 - 2
                Else 'Type Cの場合は最初のX方向の速度は0
                    evx(i) = 0
                End If
            ElseIf ef(i) = 1 Then
                '敵キャラクタ移動処理
                ney = ey(i) + evy(i) '降下
                If et(i) = 0 Then 'Type A
                     'ある程度降下したら自機に体当たりする
ようにする
                     If ney < 240 Then '画面上部に存在する
か？
                         '自機を避けるようにX方向速度の符
号を変更
                        If ex(i) > px Then
                            evx(i) = -Abs(evx(i))
                        Else
                            evx(i) = Abs(evx(i))
                        End If
                    Else
                         '自機の位置まで降下してきたときに
自機に衝突するように速度を補正
                        x = ex(i) + evx(i) * (py - ey(i))
/ evy(i)   '自機の位置まで降下してきた時のX座標を推定
                        If x > px + 8 Then evx(i) = evx(i)
- 0.5
```

➡ つぎのページに続く

リスト1の続き

```
                    If x < px - 8 Then evx(i) = evx(i) + 0.5
                    If evx(i) > 5 Then evx(i) = 5
                    If evx(i) < -5 Then evx(i) = -5
                End If
                nex = ex(i) + evx(i)
            ElseIf et(i) = 1 Then 'Type B
                '自機のX座標を目指して速度を変化させる
                xerr = (px - ex(i)) * 0.1
                If xerr > 5 Then xerr = 5
                If xerr < -5 Then xerr = -5
                nex = ex(i) + xerr
                eic(i) = eic(i) - 1 '敵弾発射間隔カウンタ更新
                If eic(i) < 1 Then '敵弾発射か？
                    '敵弾発射要求リストに項目を追加
                    ebix(ebin) = nex '敵弾初期座標
                    ebiy(ebin) = ney
                    ebivx(ebin) = 0 '真っ直ぐ降下させる
                    ebivy(ebin) = 6 '降下速度
                    ebin = ebin + 1
                    eic(i) = ei(i) '敵弾発射カウンタ初期化
                End If
            Else 'Type C
                '自機を避ける動き
                If ex(i) > px Then
                    dx = 100 / (ex(i) - px + 10)
                    If dx > 3 Then dx = 3
                    nex = ex(i) + dx
                Else
                    dx = 100 / (px - ex(i) + 10)
                    If dx > 3 Then dx = 3
                    nex = ex(i) - dx
                End If
                eic(i) = eic(i) - 1 '敵弾発射間隔カウンタ更新
                If eic(i) < 1 Then '敵弾発射か？
                    eic(i) = ei(i) '敵弾発射カウンタ初期化
                    If ney < 230 Then '発射可能なY座標か？
                        '敵弾発射要求リストに項目を追加
                        ebix(ebin) = nex '敵弾初期座標
                        ebiy(ebin) = ney
                        v = 4 '敵弾速度
                        l = distance(nex, ney, px, py) '自機までの距離を計算
                        d = l / v '何回の移動で自機まで到達するかを計算
                        ebivx(ebin) = (px - nex) / d '1回当たりの移動量を速度に設定
                        ebivy(ebin) = (py - ney) / d
                        ebin = ebin + 1
                    End If
                End If
            End If
            If nex > 556 Then nex = 556 '画面からはみ出さないようにする
            If nex < 20 Then nex = 20
            If ney > py - 8 And ney < py + 8 And nex > px - 16 And nex < px + 16 Then
                ti = ti - 100 '自機に衝突
                ef(i) = 0
            End If
            If ney > 416 Then ef(i) = 0
            putenemy ex(i), ey(i), et(i), 0
            If ef(i) = 1 Then putenemy nex, ney, et(i), col(et(i))
            ex(i) = nex '座標更新
            ey(i) = ney
        ElseIf ef(i) = 2 Then
            '敵キャラクタやられ処理
            ney = ey(i) + evy(i)
            putenemy ex(i), ey(i), et(i), RGB(0, 0, 0)
            If ney < 416 Then
                putenemy ex(i), ney, et(i), RGB(0, 0, 255)
            Else
                ef(i) = 0 '消滅
            End If
            ey(i) = ney '座標更新
        End If
    Next
    '敵弾の処理
    For i = 0 To 19
        If ebf(i) > 0 Then '存在しているか？
            nebx = ebx(i) + ebvx(i) '移動先の座標を計算
            neby = eby(i) + ebvy(i)
            If nebx < 3 Or nebx > 573 Or neby > 429 Or neby < 16 Then
                ebf(i) = 0 '画面外へ出たら消滅
            Else
                If distance(nebx, neby, px, py) < 10 Then
                    ti = ti - 100 '自機に衝突
                    ebf(i) = 0
                End If
            End If
            puteb ebx(i), eby(i), 0 '元の位置から消去
            If ebf(i) > 0 Then puteb nebx, neby, RGB(255, 255, 255) '新しい位置に描画
            ebx(i) = nebx '座標更新
            eby(i) = neby
        Else
            '存在していない場合は発生要求に応じて発生処理を行う
            If ebin > 0 Then '発生要求があるか？
                ebin = ebin - 1
                ebx(i) = ebix(ebin)
                eby(i) = ebiy(ebin)
                ebvx(i) = ebivx(ebin)
                ebvy(i) = ebivy(ebin)
                ebf(i) = 1
            End If
        End If
    Next
    ti = ti - 1
    If ti < 0 Then ti = 0
    puttime
    If ti < 1 Then gamestat = 1 '時間切れでゲームオーバー
  ElseIf gamestat = 1 Then
    With Picture1
        .CurrentX = 230
        .CurrentY = 200
        .FontSize = 15
        .ForeColor = RGB(Rnd * 255, Rnd * 255, Rnd * 255)
    End With
    Picture1.Print "Game Over"
    If keytrig3 = 1 Then
        Timer1.Interval = 0 'インターバルタイマ停止
        gameinit
    End If
  End If
End Sub

Private Sub putscore()
    Picture1.Line (200, 0)-(369, 15), 0, BF
    With Picture1
        .CurrentX = 200
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "High Score " & hiscore
    Picture1.Line (370, 0)-(469, 15), 0, BF
    With Picture1
        .CurrentX = 370
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "Score " & score
End Sub

Private Sub puttime()
    Picture1.Line (470, 0)-(575, 15), 0, BF
    With Picture1
        .CurrentX = 470
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "Time " & ti
End Sub

Private Sub scoreup(up)
    score = score + up
    If hiscore < score Then hiscore = score
    putscore
End Sub

Private Sub putplayer(x, y, c)
    Picture1.Line (x, y - 8)-(x + 16, y + 8), c
    Picture1.Line (x, y - 8)-(x - 16, y + 8), c
    Picture1.Line (x + 16, y + 8)-(x - 16, y + 8), c
End Sub

Private Sub putpb(x, y, c)
    Picture1.Line (x, y - 8)-(x, y + 8), c
End Sub

Private Sub putenemy(x, y, t, c)
    If t = 0 Then
        Picture1.Line (x + 16, y - 8)-(x - 16, y - 8), c
        Picture1.Line (x + 16, y - 8)-(x, y + 8), c
        Picture1.Line (x - 16, y - 8)-(x, y + 8), c
    ElseIf t = 1 Then
        Picture1.Line (x + 16, y)-(x + 16, y - 8), c
        Picture1.Line (x + 16, y - 8)-(x, y), c
        Picture1.Line (x, y)-(x - 16, y - 8), c
        Picture1.Line (x - 16, y - 8)-(x - 16, y), c
        Picture1.Line (x - 16, y)-(x, y + 8), c
        Picture1.Line (x, y + 8)-(x + 16, y), c
    Else
        Picture1.Line (x, y + 8)-(x + 16, y - 8), c
        Picture1.Line (x + 16, y - 8)-(x, y - 4), c
        Picture1.Line (x, y - 4)-(x - 16, y - 8), c
        Picture1.Line (x - 16, y - 8)-(x, y + 8), c
    End If
End Sub

Private Sub puteb(x, y, c)
    Picture1.Circle (x, y), 4, c
End Sub

'
'  2点間の距離を求める関数
'
Private Function distance(x0, y0, x1, y1)
    distance = Sqr((x0 - x1) * (x0 - x1) + (y0 - y1) * (y0 - y1))
End Function
```

リスト1 入力&実行方法

①VB起動

まずは，VBを起動します。新しいプロジェクトとして，「標準EXE」を選択して[開く]ボタンをクリックしてください。通常は，最初に標準EXEが選択された状態になっているので，[開く]をクリックするだけです。すると，**写真**①のような画面が表示されます。

②フォームの設定

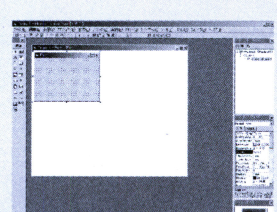
写真②のように，フォームを設定してください。プロパティは，プログラム中で更新されるので設定不要です（不用意に設定すると動作しなくなる場合があるので注意）。設定するのは，PictureBoxとTimerです。それぞれツールボックスのアイコンを選択した後に，フォーム上に貼り付けてください。貼り付ける操作は，フォーム上で左クリックした後，右下方向に少しドラッグしてから左ボタンを離せばOKです。位置，サイズなどは自由です（プログラム中で修正される）。

③リストの入力

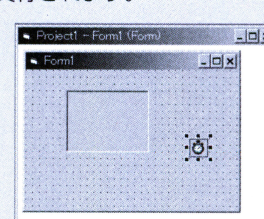
プロジェクト・ウィンドウのコード表示ボタンをクリックすると，コード・ウィンドウが表示されます。Form1のコードとして，**リスト1**を入力してください。

④実行

実行メニューから[開始]を選択すると，プログラムが実行されます。



《写真①》VBを起動したら「標準EXE」を選ぼう



《写真②》PictureBoxとTimerをフォームに配置する

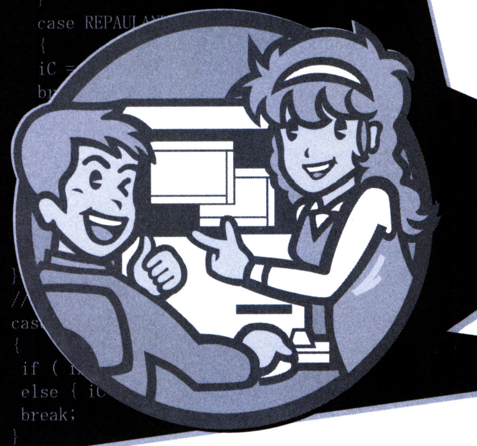
プログラマー&クリエイターへの登竜門

BM特選 投稿プログラムコーナー

Visual Basic
VB Script +DirectX
C++ Builder
HSP
Java
QBasic
ワンダーウィッチ
MUSIC

本コーナーは，さまざまな言語に対応したプログラミング学習と創作をテーマに，読者からの投稿作品で成り立っています。これから，プログラミングを勉強しようというみなさんや，プロのプログラマー／クリエイターを目指そうというみなさんにとって役立つ情報が満載です。手近にあるパソコンに入力して，お楽しみください。

それぞれの記事にはプログラムを入力／実行するときの注意点やプログラムの内容，改造方法などが紹介されていますので，よく読んでください。また，“CHECKER FLAG”というコラムがあり，デバッグ博士ことDr.Dがそれぞれのプログラムに対し批評を加えています。口が悪い人なので，かなりキビシイことも言っていますが，自分でプログラムを作ったり，これから投稿しようとしている人にとっては，とても参考になるでしょう。



プログラム投稿者へのお願い

自分の作ったプログラムを投稿しようという人はつぎのきまりを守って応募してください。

①プログラム（作品）ファイル ②プログラムの説明，住所・氏名・年齢・職業（学校名）・電話番号（連絡のつく時間も）を書いた原稿のテキスト・ファイルをフロッピーディスクなどにセーブし（プリンタで打ち出した原稿も同封のこと），お送りください。

プログラムの説明には，①内容 ②使いかた・遊びかた ③操作方法 ④プログラムのしくみ（変数表やフローチャートも）⑤プログラムの具体的入力方法，注意点 ⑥改造のアドバイス，作ったとき苦労した点 ⑦使用機種名 ⑧言語（バージョン，モードなども）などを書きます。

注1）プログラムは必ずチェックし，ミスを防ぐために同じものを2～3入れてもらえば安心です。

注2）プログラムの説明は，テキスト・ファイルをフロッピーディスクなどにセーブし，念のためプリンタで打ち出したものも同封することが基本ですが，手書き原稿でも受け付けます。その際，できるだけていねいに，マス目のある用紙を使用してください。

注3）ディスクのラベル，封筒の裏面にもプログラム名（ファイル名も），原稿ファイル名，住所，氏名，電話番号を必ず記入してください。

内容はオリジナルのもので，未発表のものに限ります。もし，参考にした記事や本がありましたら，書名，号数，記事名，筆者名，記事のタイトルなども明記しておいてください。

お送りいただいたディスクはお返しできませんので，あらかじめバックアップをとっておいてください。

掲載されたプログラムに対しては，規定の原稿料をお支払いし，掲載誌をお送りいたします。

編集部

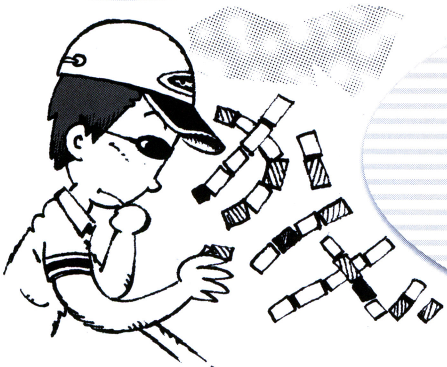
★本誌とじ込みの愛読者カードについている応募カードも忘れずに同封してください。

あて先：〒141-8715東京都品川区東五反田1-11-15　電波新聞社マイコンBASICマガジン「特選プログラム・コーナー」係

★ここに掲載されているプログラムは，ベーマガ・ホームページからもダウンロードできます。また，ホームページからの投稿も受け付けています。

http://www.basicmagazine.net/

Windows(Visual Basic V6.0)用



ハードディスク内からファイルを探せ！

Word Hunter（ワード ハンター）

吉田　真人（17歳）

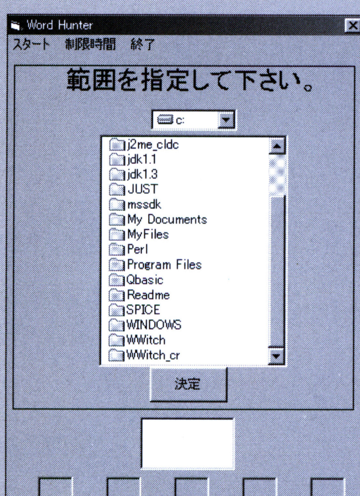
内容

パソコン上に散らばってしまった文字のかけらを集めて，何の文字かを当てるゲームです。

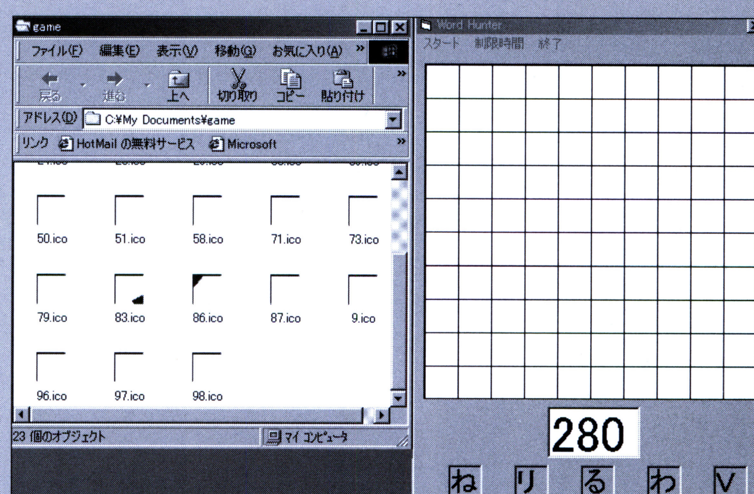
遊びかた

実行するときは，同じフォルダに“start.wav”というファイルを置いといてください。“start.wav”はC:¥WINDOWS¥MEDIAの中にありますが，もしなかったらなくてもかまいません。

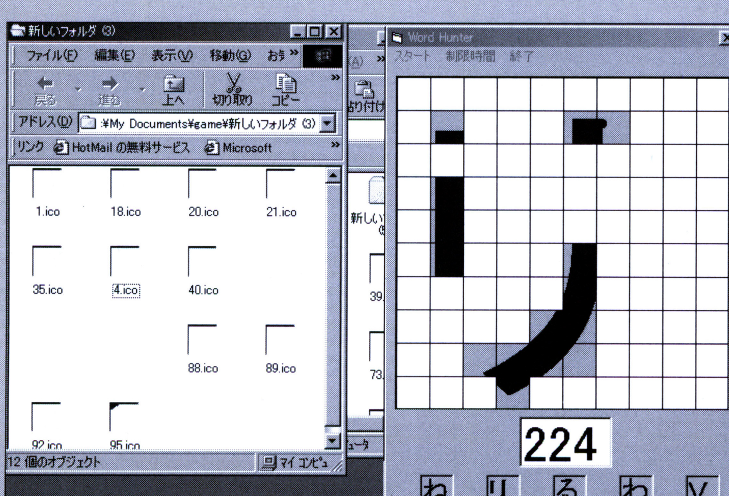
プログラムを起動したらまず制限時間を決めてください。5分，10分，15分と5分刻みで決めることができます。



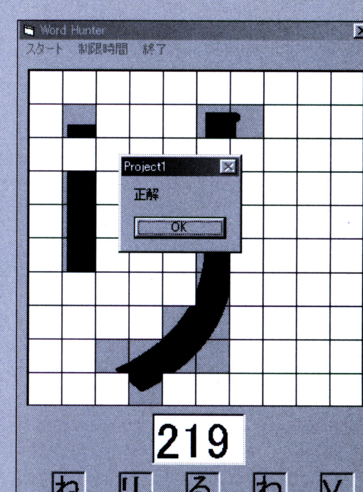
《写真1》パソコンのディスク上にパーツが散らばる



《写真2》エクスプローラなどを駆使してパーツを探そう



《写真3》ドロップすると自動的にパーツが組まれていく



《写真4》指定するフォルダによって難易度が変わるゾ

《OVER FLOW》「夏休み公開プログラミング講座」に多くの参加希望者があり，急きょ，2日間にわたって開催いたしました。プログラミングに興味のある人たちは時代が変わってもたくさんいることを再認識し，編集部もますますがんばっていくことを誓いました。

CHECKER FLAG

影：これはパソコン環境ならではのゲームだ！内容は単純だけど，自分のパソコンがゲームの舞台になる，というのがこのゲームの醍醐味と言えるね。
編：最初に実行したときは，作成されたアイコンファイルが残ったままになるのでは，という不安があったのですが，その点もしっかり対処してありました。念のためにWindowsフォルダでは遊べないようにしてあるのも，気が利いていますね。
Dr.D：このゲームの肝は，まさにこのファイル関連の処理じゃ。リストを見ると，フォルダのパス関連の命令がしっかり組まれているのがわかる。こういったシステム関連の処理は，VB側の命令ではうまく実現できないことが多いので，作品に合わせて自分で作成するのが一般的じゃろう。ない処理は自分で実現する，という姿勢はプログラムにおいて大事なことじゃな。

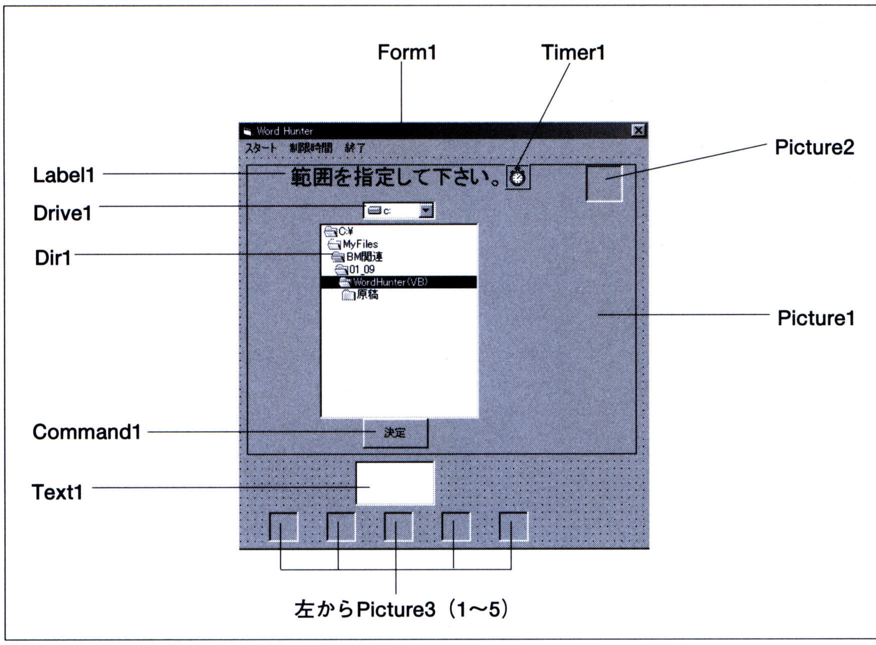
《第１表》変数表

FH()	ファイルの保存された場所
FName()	検索して見つかったフォルダの名前
A()	回答欄に表示される文字
Ti	時間のカウント用

つぎにスタートをクリックしてファイルを保存するディレクトリを指定します。ファイルは指定したディレクトリとそのサブディレクトリにランダムで保存されるので，保存されたらエクスプローラなどを開いて文字のかけらを探してください。

文字のかけらはアイコンファイルで，ファイル名は1〜100.icoとして保存されています。もし見つけることができたらそのファイルをフォームの上にドラッグ&ドロップしてください。ファイルが所定の場所に配置されます。また，ファイルが見つからないときは，見つからないマスをクリックすればファイルの保存されているパスが表示されます。

この作業を何回か繰り返していくとある文字が浮かび上がってくるので，答えがわかったら，フォームの下の回答欄の中から正解だと思うものをクリックしてください。正解ならタイムを表示して終了し，もし不正解なら残りのタイムが15秒減ってしまいます。タイムが0になるか正解を見つけたら終了します。



フォームの配置図

《第２表》メニューの設定

ｷｬﾌﾟｼｮﾝ(P)	名前(M)	ｲﾝﾃﾞｯｸｽ(X)	ﾍﾙﾌﾟｺﾝﾃｷｽﾄ番号(H)	ｼｮｰﾄｶｯﾄ(S)	ﾈｺﾞｼｴｰﾄ位置(O)	ﾁｪｯｸ(C)	有効 (E)	表示(V)	ｳｲﾝﾄﾞｳﾘｽﾄ(W)
スタート	Start		0	(なし)	0ーなし		○	○	
制限時間	Tim		0	(なし)	0ーなし		○	○	
…5分	Five		0	(なし)	0ーなし	○	○	○	
…10分	Ten		0	(なし)	0ーなし		○	○	
…15分	Fifteen		0	(なし)	0ーなし		○	○	
終了	End		0	(なし)	0ーなし		○	○	

プログラムの仕組み

パソコンのディレクトリを検索して，そこにランダムでファイルを保存しています。ただディレクトリの検索のしかたがわからないので，そのルーチンは2001年3月号のGet&Runを参考にしました。

改造のアドバイス

分割されるファイルの数を変更できるようにするとか…あとは思いつかない。

苦労した点

やはりディレクトリの検索方法です。ベーマガに載っていたのは，BMPファイルを検索するものだったのでそこから今の形にするのに苦労しました。

あとは，ファイルをドラッグ&ドロップしたときの処理です。ファイルが一つのときは簡単だったのですが，複数個のファイルをいっぺんにドロップしたときには，どうしたらよいのかわからず苦しみました。

《第３表》プロパティ表

フォームの設定	
Name	Form1激
BorderStyle	1 '固定(実線)
Caption	Word Hunter
Height	7125
Left	4065
LinkTopic	Form1
MaxButton	0 'False
MinButton	0 'False
ScaleHeight	6465
ScaleWidth	5055
StartUpPosition	2 '画面の中央
Top	2595
Width	5145
テキストボックスの設定	
Name	Text1
Font	ＭＳ Ｐゴシック
Height	735
Left	1920
Locked	-1 'True
TabIndex	11
Top	5040
Width	1335
ピクチャボックスの設定	
Name	Picture1
Appearance	0 'フラット
AutoRedraw	-1 'True
BackColor	&H80000000&
Font	ＭＳ Ｐゴシック
…Size	250
ForeColor	&H80000008&
Height	4815
Left	120
OLEDropMode	1 '手動
ScaleHeight	319
ScaleMode	3 'ピクセル
ScaleWidth	319
TabIndex	0
Top	120
Width	4815
Name	Picture2
AutoRedraw	-1 'True
AutoSize	-1 'True
Height	615
Left	5640
ScaleHeight	37
ScaleMode	3 'ピクセル
ScaleWidth	37
TabIndex	1
Top	0
Visible	0 'False
Width	615
Name	Picture3(1〜5)
AutoRedraw	-1 'True
Font	ＭＳ Ｐゴシック
Height	495
ScaleHeight	435
ScaleWidth	435
Top	5880
Width	495
Name	Picture3(1)

★7月号あくせすROOMのIWACHANさん。私も知人に1週間ほどノートパソコンを貸したら，夜中に起動する設定で返ってきて，叩き起こされたことがあります。また起動音がうるさいんだ，ウチのは。（和歌山県西牟?郡・ねこまんま18歳）……【影：ぐははは。】

★パソ破って気分いいですか？影さん。私のパソコンはよくフリーズしてしまいますが、それってパソ破の一種？（高知市・WING18歳）
……【影：気分いいわけないだろ。まわりからは非難されるし。つぐ美：影さんなりのまわりとのコミュニケーションの方法かも。】

《リスト１》フォームのプログラム・リスト

```
Option Explicit
Dim MyPath As String
Dim P$, F$, T$, H$
Dim I

Private Sub Command1_Click()
 Call Load
 Call Check
 Call Serch
 If s = True Then
  s = False
  Exit Sub
 End If
 Call Syoki2
 Call Syoki3
 Call Hozon
 Call Save
End Sub

Private Sub Fifteen_Click()
 Five.Checked = False
 Ten.Checked = False
 Fifteen.Checked = True
End Sub

Private Sub Five_Click()
 Five.Checked = True
 Ten.Checked = False
 Fifteen.Checked = False
End Sub

Private Sub form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
'終了時に残ったファイルを削除する。
 Call Del
End Sub

Private Sub End_Click()
 Call Del
 End
End Sub

Private Sub Form_Load()
 Randomize Val(Right$(Time$, 2))
 M(0) = "あいうえおかきくけこさしすせそたちつてとなにぬねのはひふへほまみむめもやゆよらりるれろわをん"
 M(1) = "アイウエオカキクケコサシスセソタチツテトナニヌネノハヒフヘホマミムメモヤユヨラリルレロワヲン"
 M(2) = "ＡＢＣＤＥＦＧＨＩＪＫＬＭＮＯＰＱＲＳＴＵＶＷＸＹＺ"
 For I = 1 To 100
  H$ = H$ + CStr(I)
 Next
 s = False
End Sub

Private Sub Picture3_Click(Index As Integer)
 If A(Index) = A(0) And s = True Then
   MsgBox "正解"
   MsgBox "あなたのタイムは" & -(Five.Checked = True) * 300 - (Ten.Checked = True) * 600 -
(Fifteen.Checked = True) * 900 - Int(Ti) & "秒です。"
   Call Syoki
 Else
  If s = True Then
   MsgBox "不正解"
   Module1.Ti = Module1.Ti - 15
  End If
 End If
End Sub

Private Sub Start_Click()
```

Index	1
Left	480
TabIndex	6
Name	Picture3(2)
Index	2
Left	1440
TabIndex	7
Name	Picture3(3)
Index	3
Left	2400
TabIndex	8
Name	Picture3(4)
Index	4
Left	3360
TabIndex	9
Name	Picture3(5)
Index	5
Left	4320
TabIndex	10
ラベルの設定	
Name	Label1
Caption	範囲を指定してください。
Font	ＭＳ Ｐゴシック
Height	495
Left	720
TabIndex	4
Top	0
Visible	0 'False
Width	'3615
コマンドボタンの設定	
Name	Command1
Caption	決定
Height	495
Left	1920
TabIndex	3
Top	4200
Visible	0 'False
Width	1095
ドライブリストボックスの設定	
Name	Drive1
Height	300
Left	1920
TabIndex	5
Top	600
Visible	0 'False
Width	1215
ディレクトリリストボックスの設定	
Name	Dir1
Height	3240
Left	1200
TabIndex	2
Top	960
Visible	0 'False
Width	2655
タイマーの設定	
Name	Timer1
Enabled	0 'False
Interval	500
Left	4440
Top	120

```
 If s = False Then
  Label1.Visible = True
  Dir1.Visible = True
  Drive1.Visible = True
  Command1.Visible = True
 Else
  If MsgBox("最初からやり直しますか？", vbYesNo) = vbNo Then
   Exit Sub
  Else
   Call Syoki
   Call Del
  End If
 End If
End Sub

Private Sub Picture1_OLEDragDrop(Data As DataObject, Effect As Long, button As Integer, shift
As Integer, X As Single, Y As Single)
 For I = 1 To Data.Files.Count
  P$ = Data.Files.Item(I)
  Call DragDrop(I, P$)
 Next
 Close
End Sub

Private Sub Picture1_MouseDown(button As Integer,shift As Integer,X As Single,Y As Single)
 If s = True Then
   MsgBox FH((Y ¥ 32) * 10 + (X + 32) ¥ 32)
 End If
End Sub

Private Sub drive1_change()
 On Error GoTo Err
 Dir1.Path = Drive1.Drive
 Exit Sub
Err:
 MsgBox Err.Description, vbCritical
End Sub

Private Sub Ten_Click()
 Five.Checked = False
 Ten.Checked = True
 Fifteen.Checked = False
End Sub
```

```
Private Sub Timer1_Timer() '制限時間のカウント
 Call Sound
 Call Timer
End Sub
```

《リスト2》モジュールのプログラム・リスト

```
Attribute VB_Name = "Module1"
Option Explicit
Option Compare Text
Private Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal X As Long, ByVal Y As
Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal xSrc As Long,
ByVal ySrc As Long, ByVal dwRop As Long) As Long
Private Declare Function sndPlaySound Lib "winmm.dll" Alias "sndPlaySoundA" (ByVal
lpszSoundName As String, ByVal uFlags As Long) As Long
Public FH(100) 'ファイルの保存された場所
Public FName() As String
Public M(2) As String
Public s As Boolean
Public A(5) As String
Public FK, Ti
Dim I, J, X, Y, MyPath
Dim P$, T$, F$
Dim FileName

Sub Syoki()  '画面の初期化
 With Form1
   .Picture1.Cls
   .Text1.Text = ""
   .Timer1.Enabled = False
   .Tim.Enabled = True
 End With
 Form1.Text1.BackColor = vbWhite
 Call Del
 For I = 1 To 5
  Form1.Picture3(I).Cls
 Next
 Form1.Picture1.BackColor = &H80000000
 s = False
End Sub

Sub Del()  '残ったファイルの削除
 If s = True Then
  For I = 1 To 100
   If Len(Dir(FH(I) + CStr(I) + ".ico", vbNormal)) <> 0 Then
    Kill FH(I) + CStr(I) + ".ico"
   End If
  Next
 Else
  For I = 1 To 100
    If Len(FH(I)) <> 0 Then
     If Dir(FH(I), vbNormal) <> "" Then
       Kill FH(I)
     End If
    End If
  Next
 End If
 Close
End Sub

Sub Syoki2()  '答えの貼り付け
 I = Int(Rnd * 3)
 If I = 2 Then
  With Form1.Picture1
    .CurrentY = 0
    .CurrentX = 30
  End With
 End If
 X = Int(Rnd * Len(M(I))) + 1
 With Form1
   .Picture1.FontSize = 245 + (I = 2) * 15
   .Picture1.Print Mid$(M(I), X, 1)
   .Picture1.Visible = False
 End With
 A(0) = Mid$(M(I), X, 1)
End Sub

Sub Syoki3() 'PIcture3(1-5)に選択肢を表示
 For I = 1 To 5
   X = Int(Rnd * 3)
   A(I) = Mid$(M(X), Int(Rnd * Len(M(X))) + 1, 1)
   Form1.Picture3(I).Print A(I)
 Next

 I = Int(Rnd * 4) + 1
 A(I) = A(0)
 With Form1
   .Picture3(I).Cls
   .Picture3(I).Print A(I)
 End With
 For I = 1 To 4
   For J = 1 To 4
     '回答欄が重複していた時の処理
     If A(I) = A(J) And I <> J Then
      X = Int(Rnd * 3)
      A(I) = Mid$(M(X), Int(Rnd * Len(M(X))) + 1, 1)
      For X = 1 To 5
       If A(I) = A(X) And X <> I Then
        While -(A(I)=A(1))-(A(I)=A(2))-(A(I)=A(3))-(A(I)=A(4))-(A(I)=A(5)) <> 1
         Y = Int(Rnd * 3)
         A(X) = Mid$(M(Y), Int(Rnd * Len(M(Y))) + 1, 1)
         With Form1
           .Picture3(X).Cls
           .Picture3(X).Print A(X)
         End With
        Wend
       End If
      Next
     End If
   Next
 Next

 With Form1.Picture2
    .ScaleWidth = 32
    .ScaleHeight = 32
 End With
 For Y = 0 To 320 Step 32
  For X = 0 To 320 Step 32
    With Form1
      .Picture1.Line (X, 0)-(X, 320)
      .Picture1.Line (0, Y)-(320, Y)
    End With
  Next
 Next
End Sub

Sub Hozon()  '画像の保存
Form1.Picture1.Line (0, 0)-(320, 320), QBColor(0), B
I = 0
For Y = 0 To 288 Step 32
 For X = 0 To 288 Step 32
   Call BitBlt(Form1.Picture2.hDC, 1, 1, 32, 32, Form1.Picture1.hDC, X, Y, &HCC0020)
   I = I + 1
   FH(I) = FName(Int(Rnd * FK))
   If Mid$(FH(I), Len(FH(I)), 1) <> "¥" Then FH(I) = FH(I) + "¥"
   SavePicture Form1.Picture2.Image, FH(I) + CStr(I) + ".ico"
   Form1.Picture2.Cls
 Next
Next
```

★父の依頼で、参議院選挙の集計用Excelマクロを作りました。たいそう集計が楽になるそうです。**(群馬県新田郡・遊牧家族18歳)**

……**【編：**さすがベーマガ読者。多方面で活躍してるね。**つぐ美：**そのうち自宅からパソコンで投票なんてことになるのかな。**】**

```
With Form1.Picture1
 .Cls
 .Visible = True
 .BackColor = vbWhite
End With

For Y = 0 To 320 Step 32
  For X = 0 To 320 Step 32
    With Form1
      .Picture1.Line (X, 0)-(X, 320)
      .Picture1.Line (0, Y)-(320, Y)
    End With
  Next
Next
Form1.Timer1.Enabled = True
s = True  'ゲームの開始フラグ
End Sub

Sub Serch()  'ディレクトリの検索
With Form1
 .Drive1.Visible = False
 .Dir1.Visible = False
 .Command1.Visible = False
 .Label1.Visible = False

 Ti=-(.Five.Checked = True)*300-(.Ten.Checked = True)*600-(.Fifteen.Checked = True)*900
 .Timer1.Enabled = False

 If Len(.Dir1.List(.Dir1.ListIndex)) = 3 Then
    MyPath = .Drive1.Drive + "¥"
 Else
    MyPath = .Dir1.List(.Dir1.ListIndex) + "¥" '検索を始めるパス
 End If
End With
 If Mid$(MyPath, 1, 10) = Form1.Drive1 + "¥WINDOWS" Then
   MsgBox "WINDOWSのシステムディレクトリは指定できません。"
   s = True
   Exit Sub
 End If
 ReDim Preserve FName(0)

 FName(0) = MyPath
 P$ = Left$(MyPath, Len(MyPath) - 1) '現在検索しているパス
 I = 0 '見つけたファイルの数
 'ループ開始
 F$ = Dir(P$ + "¥*.*", vbDirectory)
 Do
  'フォルダがあるか調べ、あればそのフォルダ内へパスを移す
  T$ = P$
  While F$ <> ""
     If (GetAttr(P$ + "¥" + F$) And vbDirectory) = vbDirectory Then
        If F$ <> "." And F$ <> ".." Then
          P$ = P$ + "¥" + F$
        End If
     End If
     If P$ = T$ Then F$ = Dir Else F$ = ""
  Wend

  If (P$ <> T$) Then
    F$ = Dir(P$ + "¥*.*", vbDirectory)
  Else
      If InStr(1, P$, "¥WINDOWS") = 0 Then
        I = I + 1
        ReDim Preserve FName(I)
        FName(I) = P$
      End If
    If P$ + "¥" <> MyPath Then
       'フォルダの階層を一つ戻る
       T$ = Mid$(P$, InStrRev(P$, "¥") + 1)
       P$ = Mid$(P$, 1, InStrRev(P$, "¥") - 1)
       F$ = Dir(P$ + "¥*.*", vbDirectory)
       '前回調べたフォルダまでdir関数の戻り値を進める
       While F$ <> T$: F$ = Dir: Wend: F$ = Dir
    Else
       Exit Do
    End If
  End If
  FK = I
Loop
End Sub

Sub DragDrop(I, P$) 'ファイルがドロップされた時の処理
  FileName = ""
  If InStr(1, P$, ".ico") <> 0 Then
     FileName = Mid$(P$, InStrRev(P$, "¥", Len(P$)) + 1, _
                Len(P$) - InStrRev(P$, "¥", Len(P$)) - 4)
     If InStr(1, P$, FileName) <> 0 And Dir$(P$, vbNormal) <> "" Then
       Form1.Picture2.Picture = LoadPicture(P$)
       Kill P$
       Call BitBlt(Form1.Picture1.hDC, -(FileName Mod 10 = 0) * 320 - 32 + (FileName Mod 10) *
32, (FileName Mod 10 = 0) * 32 + (FileName ¥ 10) * 32, 32, 32, Form1.Picture2.hDC, 1, 1,
&HCC0020)
       Form1.Picture1.Refresh
     End If
   End If
End Sub

Sub Check() '保存場所の容量チェック
 Dim fs, d, s
 Set fs = CreateObject("Scripting.FileSystemObject")
 Set d = fs.GetDrive(fs.GetDriveName(Form1.Drive1.Drive))
 If FormatNumber(d.FreeSpace / 1024, 0) < 4.09 * 100 Then
   MsgBox "ディスク容量が足りません！"
   End
 End If
End Sub

Sub Sound()  'チクタク音の再生
 If Dir("Start.wav", vbNormal) <> "" Then
  Call sndPlaySound("Start.wav", 0) 'チクタク音の再生
 End If
End Sub

Sub Timer() '制限時間のカウント
 Ti = Ti - 0.5
 Form1.Text1.Text = Int(Ti)
 If Ti <= 0 Then
   MsgBox "残念、また挑戦してくださいね。"
   Call Syoki
 End If
End Sub

Sub Load()
If Dir("File.log", vbNormal) <> "" Then
 Open "File.log" For Input As #1
 For I = 1 To 100
  Input #1, FH(I)
 Next
 Call Del
 Close
End If
End Sub

Sub Save()
Open "File.log" For Output As #1
For I = 1 To 100
 Print #1, FH(I) + CStr(I) + ".ico"
Next
Close
End Sub
```

★学校ではみんなケータイ持ってます。暇さえあればメールチェック！電話というよりメール機になってます。（埼玉県北葛飾郡・ブラウン16歳）……**【影：**目に浮かぶ光景だ。（ピンポーン）…おっ、メールだ。なになに。**つぐ美：**勝手に私のメール見ないでよ。**】**

Windows(Visual Basic V6.0)用



UFOの攻撃をかわし続けろ

Sky Fight

スカイ　ファイト

宮尾　歩（13歳）

内容

UFOが最新の人工衛星に攻撃をしかけてきました。コンピュータの指示に従って安全ゾーンに逃げ込まないと人工衛星が爆破されてしまいます。そこで君が人工衛星の操作を命じられました。

さて，人工衛星はいつまで持つでしょうか？

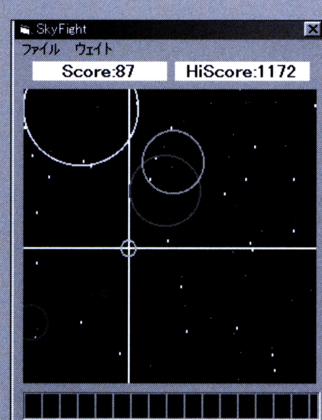
使いかた・遊びかた

水色の人工衛星を黄色のゾーンにくぐらせていきます。赤ゾーンはUFOの攻撃で，ダメージを受けてしまうのでよけてください。緑ゾーンは点数10点UPですが，非常にくぐりにくいのであまり意識しないほうがいいかもしれません。

ダメージを受けると画面が赤く光ります。人工衛星の耐久力がなくなったらゲーム・オーバーです。なるべく大きいスコアを出しましょう。



《写真1》中学生が作成したVBゲームだ！！



《写真2》一見、シューティング・ゲームみたいだが



《写真3》安全な輪の中へ移動していくアクション・ゲーム



《写真4》長くプレイするほど得点が高くなる

操作方法

ゲームを開始させるには，メニューからスタートを選ぶか，[F5]キーを押してください。カーソル・キーで移動します。ゲームを終了するときは，[F4]キーで終わります。

プログラムについて

ループで敵の処理などをまとめてあります。特徴としては，Pentium 100MHz以下でも快適に動きます　入力方法は，フォームとモジュールを用意してプロパティを設定して入力するだけです。

CHECKER FLAG

つぐ美：VBのグラフィック描画命令で画面を描いているので，見た目は単調ですが，ゲームとしてのバランスがしっかりしている作品です。友達のアドバイスによって改良を重ねたというのが伝わってきますね。

編：ファイルによるハイスコア記録もあり，ゲームとしての基本的な要素はほとんど持っている作品です。これからの投稿作品にも期待できます。

Dr.D：リストにきめ細かくコメントが付いているのがありがたいのぉ。細かい点では，ゾーンの描画は大きさの変更だけでなく，同時に色も変更させているので，より奥行きを感じる動きになっておる。

影：少し改造すればシューティング・ゲーム形式にもできそうな作品だ。

★今，このハガキ，朝に学校で書いています。窓から好きな人が登校してくるのが見えました。萌！（北海道天塩町・ゆきちゃんLOVE15歳）……【影：おーっ，青春してるねぇ。私の学生時代を思い出してしまったよ。つぐ美：モテナカッタ時代ね。影：失礼な！】

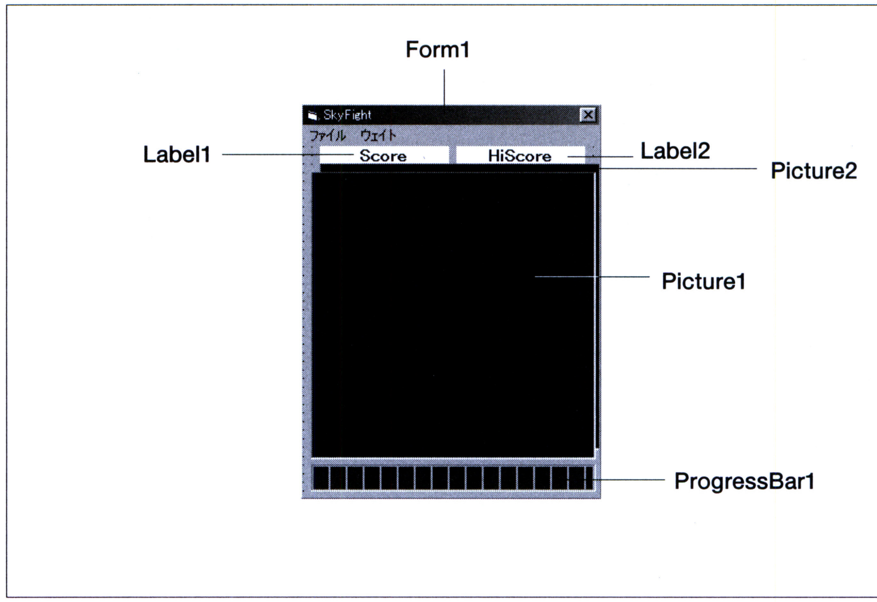
苦労点と改造のアドバイス

やっぱり敵の処理でした。変なのは，キーイベント（KeyPreview）をFalseに設定したままで敵が動くのに，キャラクターが動かないので，原因を調べるのに3日ぐらいかかってしまいました。

基本的な部分しか作成していないので，改造は音を付けるとか，3D効果をつけるなど，いろいろできると思います。

最後に

僕が小学生のときに作ったものを,VB版にリメイクしたもの。友達にもいろいろ言われて改良に改良を重ねました。



フォームの配置図

《第１表》プロパティ表

フォームの設定	
Name	Form1
BorderStyle	1 '固定(実線)
Caption	SkyFight
Height	5550
Left	1500
LinkTopic	Form1
MaxButton	0 'False
MinButton	0 'False
ScaleHeight	555
ScaleMode	0 'ユーザー
ScaleWidth	425.35
Top	585
Width	4260
ピクチャボックスの設定	
Name	Picture1
AutoRedraw	-1 'True
AutoSize	-1 'True
BackColor	&H00000000&
DrawWidth	2
Height	453.988
Left	12.24
ScaleHeight	400
ScaleMode	0 'ユーザー
ScaleWidth	400
TabIndex	1
Top	40.859
Visible	0 'False
Width	408.01
Name	Picture2
AutoRedraw	-1 'True
AutoSize	-1 'True
BackColor	&H00000000&
ForeColor	&H0000FFFF&
Height	453.988
Left	24.481
ScaleHeight	400
ScaleMode	0 'ユーザー
ScaleWidth	400
TabIndex	0
Top	27.239
Visible	0 'False
Width	408.01
ラベルの設定	
Name	Label1
Alignment	2 '中央揃え
BackColor	&H00FFFFFF&
Caption	Score
Font	ＭＳ Ｐゴシック
ForeColor	&H00000000&
Height	28.942
Left	24.481
TabIndex	3
Top	0
Width	185.134
Name	Label2
Alignment	2 '中央揃え
BackColor	&H00FFFFFF&
Caption	HiScore
Font	ＭＳ Ｐゴシック
ForeColor	&H00000000&
Height	28.942
Left	220.325
TabIndex	4
Top	0
Width	185.134
プログレスバーの設定	
Name	ProgressBar1
Appearance	1
Height	42.561
Left	12.24
TabIndex	2
Top	503.926
Width	408.52

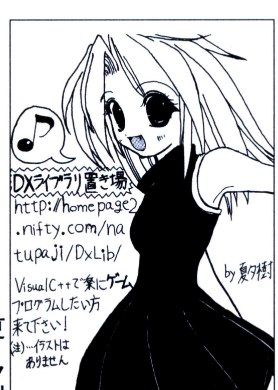
《第２表》メニューの設定

ｷｬﾌﾟｼｮﾝ(P)	名前(M)	ｲﾝﾃﾞｯｸｽ(X)	ﾍﾙﾌﾟｺﾝﾃｷｽﾄ番号(H)	ｼｮｰﾄｶｯﾄ(S)	ﾈｺﾞｼｴｰﾄ位置(O)	ﾁｪｯｸ(C)	有効（E)	表示(V)	ｳｲﾝﾄﾞｳﾘｽﾄ(W)
ファイル	M_File		0	(なし)	0 ーなし		○	○	
…ゲーム開始	M_GameStart		0	F5	0 ーなし		○	○	
…-	M_Hr		0	(なし)	0 ーなし		○	○	
…終了	M_End		0	F4	0 ーなし		○	○	
ウェイト	M_Weit		0	(なし)	0 ーなし		○	○	



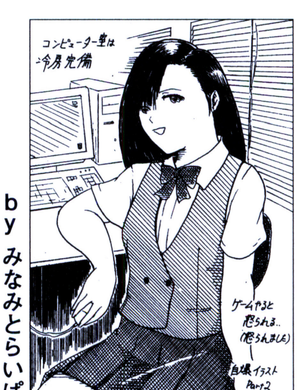
by 都田 祥

評…ということです。今後まちがわないように！（編）



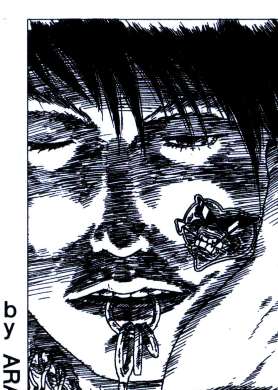
by 夏 夕樹

評…ホームページ確認しました。これは大丈夫です（編）



by みなみとらいぱ

評…たしかに，真夏はパソコン室に群がるのが一番！（影）



by ARAI

評…なんか怖いわね。何をイメージしてるんでしょう（つぐ美）

★キャッチャー商品というのはミットとか，マスクとかのことですか？同ネタ多数。（岩手県釜石市・**鉄人坂井23歳**）……【**影**：プロテクタも添えてね。**編**：読者のみんなが本気にしたらどうするんですか。**影**：誰も私の言うことなんて信じてないようだし。グスン。】

《リスト１》フォームのプログラム・リスト

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
Select Case KeyCode 'キャラクタキー判定
    Case vbKeyRight
        R = True
    Case vbKeyLeft
        L = True
    Case vbKeyUp
        U = True
    Case vbKeyDown
        D = True
End Select
End Sub

Private Sub Form_KeyUp(KeyCode As Integer, Shift As Integer)
Select Case KeyCode 'キャラクタキー判定
    Case vbKeyRight
        R = False
    Case vbKeyLeft
        L = False
    Case vbKeyUp
        U = False
    Case vbKeyDown
        D = False
End Select
End Sub

Private Sub Form_Load()
On Error GoTo Err 'エラートラップ
    Randomize '乱数列を初期化
    Wt = 25 'ウェイト初期値設定
    Call PrintInp(90, 160, 28, RGB(0, 255, 0), "SkyFight") 'タイトル表示
    Open App.Path & "¥HiScore.dat" For Input As #1 'スコア読み込み
        Input #1, HiScore
    Close #1
Err:
    Label2.Caption = "HiScore:" & HiScore 'ハイスコア表示
End Sub

Private Sub Form_Paint() '画面
    Call BitBlt(Form1.hDC, 8, 25, 400, 400, Picture1.hDC, 0, 0, vbSrcCopy)
End Sub

Private Sub Form_Unload(Cancel As Integer)
    EndFlag = True 'ループをぬける
End Sub

Private Sub M_End_Click()
    Unload Form1 'アプリケーション終了
End Sub

Private Sub M_GameStart_Click()
    Call GameInit 'ゲーム初期化ルーチン
    Call Game 'メインループ
End Sub

Private Sub M_Weit_Click() 'ウェイト設定
On Error Resume Next
    A = InputBox("ウェイトを入力して下さい。0～1000", "ウェイト設定", Wt)
    If Err = 0 Then
        If A < 0 Or A > 1000 Then '有効範囲以外
            Exit Sub
        End If
        Wt = A '値を設定
    End If
End Sub
```

《リスト２》モジュールのプログラム・リスト

```
Attribute VB_Name = "Module1"
Option Explicit '変数宣言強制

'ＡＰＩ宣言
Public Declare Function timeGetTime Lib "winmm.dll" () As Long
```

★ベスト・プログラムの解説はすばらしいことです。続けてください。（福井県武生市・**つかち～。16歳**）……【**Dr.D**：新連載に多くの意見をもらっておる。うれしい限りじゃ。ワシもDr.Jに負けずにがんばらんとな。ファンレターも待っておるぞ。よろしく！】

```
Public Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal X As Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal xSrc As Long, ByVal ySrc As Long, ByVal dwRop As Long) As Long

'変数宣言
Public Wt, Time As Long 'ウェイト用
Public Score As Integer 'スコア
Public PX, PY, Pw As Integer '自キャラ座標
Public Ld(4), Lu(4), Big(4), Color(4) As Integer '敵キャラ座標
Public R, L, U, D As Boolean 'キーフラッグ
Public EndFlag As Boolean 'ゲーム終了フラッグ
Public HiScore As Integer 'ハイスコア
Public A, B As Integer 'カウンター雑用

Public Sub Game() 'メインプロシージャ
    Do Until EndFlag = True 'ループ
        DoEvents '他イベント監視
        Time = timeGetTime 'タイム取得
        Score = Score + 1 'スコア加算
        Form1.Label1.Caption = "Score:" & Score 'スコア表示
        '背景を写す（クリア）
        Call BitBlt(Form1.Picture1.hDC,0,0,400,400,Form1.Picture2.hDC,0,0,vbSrcCopy)
        '***************キャラクタ制御******************
        If R = True Then '右
            PX = PX + 8
        End If
        If L = True Then '左
            PX = PX - 8
        End If
        If D = True Then '下
            PY = PY + 8
        End If
        If U = True Then '上
            PY = PY - 8
        End If
        If PX < 5 Then PX = 10 '移動範囲設定
        If PX > 395 Then PX = 390
        If PY < 5 Then PY = 10
        If PY > 395 Then PY = 390
        Form1.Picture1.Line (PX, 0)-(PX, 400), RGB(255, 255, 255) '自キャラを描く
        Form1.Picture1.Line (0, PY)-(400, PY), RGB(255, 255, 255)
        Form1.Picture1.Circle (PX, PY), 10, RGB(0, 255, 255)
        '*******************敵キャラクター制御****************
        For A = 0 To 4 '伽羅の数だけループ
            Select Case A '処理分岐
            Case 4 '安全ゾーン（黄色）
                Call LoopMove(A, 90)
                '黄ループを描く
                Form1.Picture1.Circle (Lu(A), Ld(A)), Big(A), RGB(Color(A), Color(A), 0)
                If Big(A) = 80 Then '衝突判定
                    If Hit(A, 80) = False Then
                        Call Hp_out(A) '体力減少
                    End If
                End If
            Case 3 'アイテム 10点ＵＰ（緑色）
                Call LoopMove(A, 40)
                Form1.Picture1.Circle (Lu(A), Ld(A)), Big(A), RGB(0, Color(A), 0)'緑ループ
          If Big(A) = 30 Then
                If Hit(A, 30) = True Then '衝突判定
                    Score = Score + 10 'スコア加算
                    Form1.Label1.Caption = "Score:" & Score 'スコア表示
                End If
            Case Else '障害物（赤色）
                Call LoopMove(A, 60)
                If Big(A) = 50 Then '衝突判定
                    If Hit(A, 50) = True Then
                        Call Hp_out(A) '体力減少
                    End If
                End If
                Form1.Picture1.Circle (Lu(A), Ld(A)), Big(A), RGB(Color(A), 0, 0)'赤ループ
            End Select
        Next A 'ループ終わり
        'フォームに画像を転送
        A = BitBlt(Form1.hDC, 8, 25, 400, 400, Form1.Picture1.hDC, 0, 0, vbSrcCopy)
        While (Time + Wt + 25 > timeGetTime): DoEvents: Wend 'ウェイト
    Loop 'ループ
End Sub
```

★あっつ～い～。あっつ～い～。ぐでー，影さ～ん，カキ氷食べた～い。（埼玉県さいたま市・**グダー16歳**）……【**影**：なんかこんなハガキが多いぞ。やる気があるのか！私は毎日，熱でこもったパソコンルームで，投稿チェックを……。**編**：やってほしいもんです。】

```
Public Sub GameInit() 'ゲームリセット
    'ゲームリセット
    EndFlag = False 'ゲーム終了フラッグオフ
    Form1.Picture1.Cls '裏画面リセット
    Score = 0 'スコアリセット
    For A = 0 To 3 'ループ待機
        Lu(A) = -100
    Next A
    For A = 0 To 60 Step 20 'ループリセット
        Big(A / 20) = A
    Next A
    Big(4) = 90
    Form1.KeyPreview = True 'キーイベント認識
    PX = 200: PY = 200: Pw = 100 'プレイヤー初期化
    Form1.ProgressBar1.Value = Pw 'パワーケージ初期化
    Form1.Picture2.Cls '背景クリア
    For A = 0 To 50 '背景の☆描画
        B = Int(Rnd * 2 + 1) '☆サイズ
        Form1.Picture2.DrawWidth = B
        '星をランダムに描く
        Form1.Picture2.PSet (Int(Rnd * 400), Int(Rnd * 400)), RGB(255, 255, 255)
    Next A
    R = False: L = False: U = False: D = False 'キーリセット
End Sub

Public Function Hit(F, H) As Boolean 'プレイヤーとの衝突判定
    If Sqr((PX - Lu(F)) ^ 2 + (PY - Ld(F)) ^ 2) < 10 + H Then
        Hit = True '衝突している
    Else
        Hit = False '衝突していない
    End If
End Function

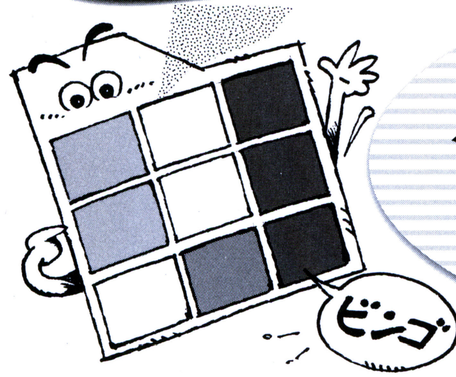
Public Sub PrintInp(PrX, Pry, Prs, Prcol, Pr As String) '画面に文字を描く
    Form1.Picture1.FontSize = Prs 'フォントサイズ
    Form1.Picture1.CurrentX = PrX 'フォントX座標
    Form1.Picture1.CurrentY = Pry 'フォントY座標
    Form1.Picture1.ForeColor = Prcol 'フォントカラー
    Form1.Picture1.Print Pr '文字
End Sub

Public Sub LoopMove(A, B) '敵を動かす
    If Big(A) > B Then '敵新規作成
        Lu(A) = Int(381 * Rnd + 10) 'でたらめな位置に
        Ld(A) = Int(381 * Rnd + 10)
        Big(A) = 0 '大さ初期化
        Color(A) = 135 '色を薄く設定
    Else '敵移動中
        Big(A) = Big(A) + 2 '大さプラス
        Color(A) = Color(A) + 5 '色を濃くする
    End If
End Sub

Public Sub Hp_out(B)  'ダメージをくらったとき
    Form1.Picture1.Line (0, 0)-(400, 400), RGB(255, 0, 0), BF '画面を赤く光らせる
    Pw = Pw - 20 '体力減少
    Form1.ProgressBar1.Value = Pw 'パワーケージに値を入れる
    Big(B) = 100 '大きさをとりあえず１００に設定
    If Pw = 0 Then '体力が0になったとき
        Form1.KeyPreview = False 'キーイベント認識停止
        Call PrintInp(50, 180, 28, RGB(0, 255, 255), "GAME OVER") '文字表示
        '画面をフォームに転送
        Call BitBlt(Form1.hDC, 8, 25, 400, 400, Form1.Picture1.hDC, 0, 0, vbSrcCopy)
        If Score > HiScore Then 'スコアがハイスコアを上回ったとき
            HiScore = Score 'ハイスコアにスコアを入れる
            Open App.Path & "¥HiScore.dat" For Output As #1 'ハイスコア保存
                Write #1, HiScore '書きこむ
            Close #1 'ファイルをクローズ
            Form1.Label2.Caption = "HiScore:" & HiScore 'ハイスコアを表示させる
            Call PrintInp(300, 370, 14, RGB(255, 255, 0), "HiScore") 'ハイスコアを知らせる
        End If
        EndFlag = True 'エンドフラッグオン
    End If
End Sub
```

★ホームページが着実に完成へと近づいています。イエーイ。でもここで問題が！！制作費が！ということでどうしましょうか。(愛知県刈谷市・**ゆうぐれ13歳**)……**【影：それは困った。じつは私のホームページも資金が足りずにストップしてるんだよ。くそー。】**

Windows(Visual Basic V6.0)用



色がそろった列を瞬時に判断

ビンゴ？ゲーム

小薗　直哉（15歳）

内容

3×3のますの色がそろったらボタンを押すゲームです。

使いかたと遊びかた

3×3のますの下もしくは右に！ボタンがあります。列がそろったらそれを押してください。レベルに10を掛けたものが得点として加算され，レベルに5を掛けたものが失点として減点されます。

3回成功するごとにレベルが1上がります。3回失敗するごとにレベルが1下がります。レベルが1上がると変化の速度が2倍になります。レベルは1から5まであります。

三つそろうのに時間がかかるので，色の付いたボタンをクリックすると色が変わるようにしました。簡単にそれを使わないよう１回につき－５点となってます。

tabindexのふりかたを工夫しました。矢印キーでボタンを選び，スペース・キーやエンター・キーで打つとおもしろいと思います。

注意

command1(1から9)のstyleは，グラフィクスにしないとカラフルなゲームが楽しめません。というよりゲームになりません。

作者からのコメント

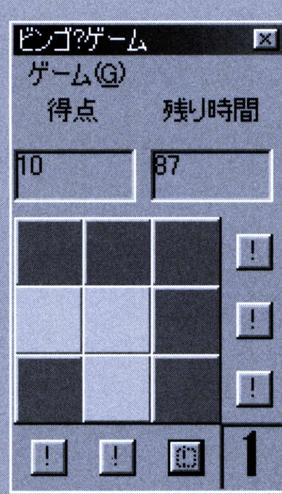
時間体系と点数体系が悪いかもしれません。

《第１表》メニューの設定

ｷｬﾌﾟｼｮﾝ(P)	名前(M)	ｲﾝﾃﾞｯｸｽ(X)	ﾍﾙﾌﾟｺﾝﾃｷｽﾄ番号(H)	ｼｮｰﾄｶｯﾄ(S)	ﾈｺﾞｼｴｰﾄ位置(O)	ﾁｪｯｸ(C)	有効(E)	表示(V)	ｳｲﾝﾄﾞｳﾘｽﾄ(W)
ゲーム(&G)	game		0	(なし)	0－なし		○	○	
開始(&S)	start		0	F2	0－なし		○	○	
説明(&H)	howtouse		0	(なし)	0－なし		○	○	
….-	line		0	(なし)	0－なし		○	○	
…終了(&X)	exit		0	(なし)	0－なし		○	○	



《写真１》ビンゴのルールを利用したゲーム



《写真２》色が揃った列を選択していこう

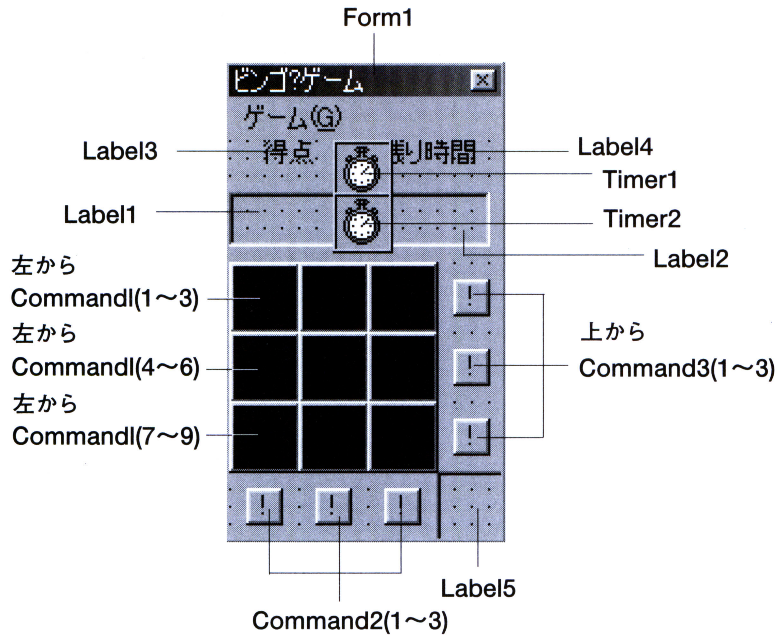


《写真３》レベルが上がると色の変更時間も速くなる



《写真４》制限時間内で高得点を目指せ

★「めざましテレビ」のスポーツコーナーのBGMって，『グラⅢ』の曲に聞こえる。「ジャッジャッジャツン」とオーケストラヒット鳴るし。(鹿児島県川内市・純太6641 27歳)……【つぐ美：するどいかも。私も朝は「めざましテレビ」派よ。影：テレビなーい。】



フォームの配置図

CHECKER FLAG

つぐ美：単純なゲームなので，遊びやすいですね。得点によるスピードアップがあるので，遊び始めるとけっこうハマりますよ。

編：VBでゲームを作りたいと思っている読者は，いきなりこだわったゲームを作らず，こういった単純なゲームからゲームプログラムのコツを覚えていくのがオススメですね。

Dr.D：リスト面でも非常に単純な命令のみで実現しておるので，初心者には参考になる作品じゃ。コマンドボタンを配列で宣言することで，この手のゲームに付きものの仮想画面用の配列が必要ないのが特徴じゃ。

《第3表》変数表

limit	制限時間
revel	レベル
pts	得点
success	成功回数
miss	失敗回数

《第2表》プロパティ表

フォームの設定	
Name	Form1
BorderStyle	4 '固定ツールウィンドウ
Caption	ビンゴ?ゲーム
Height	3345
Left	105
LinkTopic	Form1
MaxButton	0 'False
MinButton	0 'False
ScaleHeight	2745
ScaleWidth	1890
ShowInTaskbar	0 'False
StartUpPosition	3 'Windows の既定値
Top	105
Width	1980
コマンドボタンの設定	
Name	Command1(1～9)
BackColor	&H00000000&
Height	495
Style	1 'グラフィックス
Width	495
Name	Command1(1)
Index	1
Left	0
TabIndex	6
Top	840
Name	Command1(2)
Index	2
Left	480
TabIndex	7
Top	840
Name	Command1(3)
Index	3
Left	960
TabIndex	8
Top	840
Name	Command1(4)
Index	4
Left	0
TabIndex	9
Top	1320
Name	Command1(5)
Index	5
Left	480
TabIndex	10
Top	1320
Name	Command1(6)
Index	6
Left	960
TabIndex	11
Top	1320
Name	Command1(7)
Index	7
Left	0
TabIndex	12
Top	1800
Name	Command1(8)
Index	8
Left	480
TabIndex	13
Top	1800
Name	Command1(9)
Index	9
Left	960
TabIndex	14
Top	1800
Name	Command2(1～3)
Caption	!
Height	255
Top	2400
Width	255
Name	Command2(1)
Index	1
Left	120
TabIndex	0
Name	Command2(2)
Index	2
Left	600
TabIndex	1
Name	Command2(3)
Index	3
Left	1080
TabIndex	2
Name	Command3(1～3)
Caption	!
Height	255
Left	1560
Width	255
Name	Command3(1)
Height	255
Index	1
TabIndex	5
Top	960
Name	Command3(2)
Index	2
TabIndex	4
Top	1440
Name	Command3(3)
Index	3
TabIndex	3
Top	1920
ラベルの設定	
Name	Label1
BackStyle	0 '透明
BorderStyle	1 '実線
Height	375
Left	0
TabIndex	15
Top	360
Width	855
Name	Label2
BackStyle	0 '透明
BorderStyle	1 '実線
Height	375
Left	960
TabIndex	16
Top	360
Width	855
Name	Label3
Alignment	2 '中央揃え
BackStyle	0 '透明
Caption	得点
Height	255
Left	0
TabIndex	17
Top	0
Width	855
Name	Label4
Alignment	2 '中央揃え
BackStyle	0 '透明
Caption	残り時間
Height	255
Left	960
TabIndex	18
Top	0
Width	855
Name	Label5
Alignment	2 '中央揃え
BackStyle	0 '透明
BorderStyle	1 '実線
Font	MS ゴシック
Height	495
Left	1440
TabIndex	19
Top	2280
Width	495
タイマーの設定	
Name	Timer1
Enabled	0 'False
Interval	1000
Left	720
Top	0
Name	Timer2
Enabled	0 'False
Interval	1000
Left	720
Top	360

★HPめぐりって楽しいですね。Z氏とかN氏のとか。新しいサイトも訪ねてみたいですね。（青森県南津軽郡・**ていこく22歳**）……【**影**：岬めぐりって知ってる？**Dr.D**：ワシも好きじゃ，その歌。**影**：いいですよねぇ。**つぐ美**：2人で話がはずんでる姿，初めて見た！】

『ビンゴ？ゲーム』のプログラム・リスト

```
Dim limit As Integer
Dim pts As Integer
Dim revel As Integer
Dim success As Integer
Dim miss As Integer
Dim messe
Private Sub form_load()
messe = "色がそろったらボタンを押すゲームです。"
End Sub
Private Sub start_Click()
Timer1.Enabled = False
Timer2.Enabled = False
Randomize Timer
pts = 0
revel = 1
Label1.Caption = CStr(pts)
Label5.Caption = CStr(revel)
success = 0
miss = 0
Label2.Caption = "準備中"
ChangeAll
limit = 90
Timer1.Enabled = True
Timer2.Enabled = True
Label2.Caption = CStr(limit)
End Sub

Private Sub howtouse_Click()
MsgBox messe, vbInformation, Form1.Caption
End Sub

Private Sub exit_Click()
End
End Sub

Private Sub Command1_Click(Index As Integer)
If Timer1.Enabled = False Then Exit Sub
changeone (Index)
pts = pts - 5
Label1.Caption = CStr(pts)

End Sub

Private Sub Command2_Click(Index As Integer)
If Timer1.Enabled = False Then Exit Sub

If Command1(Index).Tag = Command1(Index + 3).Tag Then
If Command1(Index + 3).Tag = Command1(Index + 6).Tag Then
getpts
Else: lostpts
End If
Else: lostpts
End If

End Sub

Private Sub Command3_Click(Index As Integer)
If Timer1.Enabled = False Then Exit Sub

If Command1(Index * 3 - 2).Tag = Command1(Index * 3 - 1).Tag Then
If Command1(Index * 3 - 1).Tag = Command1(Index * 3).Tag Then
getpts
Else: lostpts
End If
Else: lostpts
End If

End Sub

Private Sub changeone(b As Integer)
a = Int(Rnd * 3)

Select Case a
Case 0: Command1(b).BackColor = vbRed
Case 1: Command1(b).BackColor = vbGreen
Case 2: Command1(b).BackColor = vbCyan
End Select
Command1(b).Tag = Command1(b).BackColor

End Sub

Private Sub ChangeAll()
For c = 1 To 9
changeone (c)
Next c
End Sub

Private Sub getpts()
pts = pts + 10 * revel
Label1.Caption = CStr(pts)

success = success + 1
If success = 3 Then revel = revel + 1: success = 0
If revel > 5 Then revel = 5: success = 0
Label5.Caption = CStr(revel)
End Sub

Private Sub lostpts()
pts = pts - 5 * revel
Label1.Caption = CStr(pts)

miss = miss + 1
If miss = 3 Then revel = revel - 1: miss = 0
If revel < 1 Then revel = 1: miss = 0
Label5.Caption = CStr(revel)
End Sub

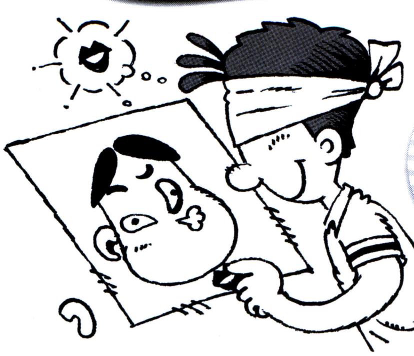
Private Sub Timer1_Timer()
ChangeAll
Timer1.Interval = 250 * (2 ^ (4 - revel))
End Sub

Private Sub Timer2_Timer()
limit = limit - 1
Label2.Caption = CStr(limit)
If limit = 0 Then
Timer1.Enabled = False
Timer2.Enabled = False
MsgBox "時間切れです。", vbExclamation, Form1.Caption
Label2.Caption = "時間切れ"
End If

End Sub
```

★最近は豆乳が美味しいのです。メーカーは●文。みなさんもいかがですか。（北海道札幌市・春陽19歳）……【つぐ美：身体に良さそうね。チャレンジしてみるわ。それよりハガキのすみに書いてあった「ドクペはあれ以来口にしていません」というのが気になるー。】

Windows(Visual Basic V6.0)用



好きな写真で福笑いが楽しめる

FUKUWARAI 2
（ふくわらい　ツー）

M.Hayashi（51歳）

内容

「福笑い」をパソコンで遊べるようにしたゲームです。ベーマガ2000年９月号の「今月はもう一歩でしたで賞」で取り上げていただいた『FUKUWARAI』の再投稿です。用意されている顔の他に，簡単に自分の顔・友達の顔などを使って「福笑い」ができるようになっています。

遊びかた

まず，「顔選択」ボタンを押して好きな顔を選んでください。つぎに，「練習」モードと「本番」モードを選んでください。「練習」モードは目・鼻・口などの各部分が消えません。「本番」モードは各部分が消えます。そして，各部分を顔の上へドラッグ&ドロップで移動してください。詳しくは，画面下部に表示される説明を見てください。

顔の作成・追加のしかた

顔の作りかたには，2とおりの方法があります。ひとつ目はメニューの「ゲーム」-「顔の作成」から作る方法です。これはデジカメで撮影した顔などの画像ファイルさえあれば簡単に作成できます。

もうひとつは，ペイントソフトなどで顔と顔の各部分をあらかじめ作っておいて，メニューの「ゲーム」-「顔の追加」で読み込む方法です。こちらは少し手間がかかりますが，何個でも顔を作成できて，自分で納得いく顔を作れます。

詳しい説明はダウンロードファイルに収録してあるヘルプを読んでください。

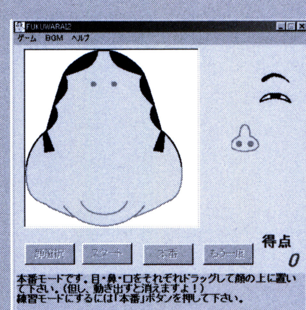
改造点や苦労点など

顔の作成で，塗りつぶしの色をより自然な色にするには，どうしたらいいですか？このプログラムでは，psetメソッドで２色を交互に塗っていますが，この色数を大きくすればいいのでしょうか？

前回投稿したものとおなじ部分もかなりあって，長いリストになってしまいました。もっともっと短くすることができるように「日曜プログラマー」の腕をみがいていきたいと思います。また音楽の再生については，過去のベーマガを全面的に参考にさせていただきました。



《写真1》21世紀も福笑いは健在だ



《写真2》目隠しの代わりにパーツが消える仕組み



《写真3》この出来上がりの顔が福笑いの醍醐味



《写真4》好きな画像を福笑いの題材にできるのが特徴

CHECKER FLAG

編：ゲーム内容は前作とほとんど同じだったのですが，オリジナルの顔を作成する機能を評価しての採用です。画像ファイルさえあれば作成できるモードがいいですね。

Dr.D：塗りつぶしについての質問じゃが，自然な塗りつぶしというのはかなり難しい部類に入るゾ。本格的に作成するには，グラフィックの知識だけでなく，画像処理の知識も必要になる。塗りかただけでなく，パーツを切り取った後の画像認識もしなければならないからじゃ。

つぐ美：できた顔に対する採点があるのも特徴ですね。たまに採点された点数に文句をつけたくなるときもあります。また，あまりに変な形だとマイナスの点が表示されてしまったりしますね。

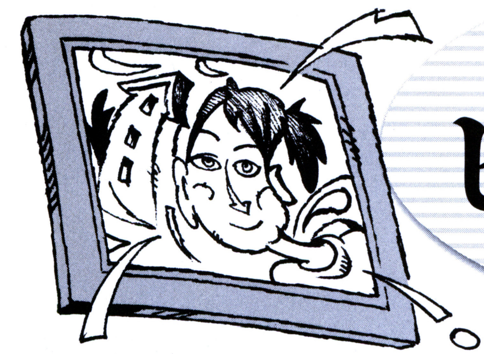
★コンビニにて，ドクターペッパー（500ml）を発見。…どうする？（**鈴木真理子21歳**）……【**編**：①即座に買う。②じーっと考えこむ。③他人の動きを見る。④コカコーラを買う。⑤ビールを買う。**影**：私は⑤だな。**編**：ゲーム・オーバー。**影**：なんじゃそれ。】

(超大作プログラム)　リストが誌面に掲載できない作品です。
BMホームページ(http://www.basicmagazine.net)からダウンロードしてお楽しみください。

Windows(VBScript+DirectX)用

3Dモデル上で特殊効果画像

ピクチャVer.1.04



金井　義男(39歳)

内容

2Dトランスフォームによる特殊効果画像表示をするプログラムです。画像および，3DモデルをDirectAnimationの2Dトランスフォームによる特殊効果（29種類）と組み合わせて表示します。

使いかた・遊びかた

画像やXファイルを読みこみ，トランスフォームの変更や位置移動，拡大，縮小，透明度，適用時間などの変更ができます。出来上がったイメージは，キャプチャ（PrtScキー）でクリップボードにbmp形式で保存したものをファイル化してホームページなどで使用します。

操作方法

マウスの右クリックと左クリックを使って，トランスフォームを切り替えます。メニュー表示されたキーボタンで各種設定値を変更します。

プログラムの入力法と注意点

メモ帳を使って入力，改造ができますが，デバッガとテキストエディタ機能のあるホームページ作成ツールを使用することをお薦めします。

改造のアドバイスと苦労点

まだまだ理解できてないことが多くて苦労しています。また，ボタン型の3Dモデルを自作して，画像を貼り付け，ホームページで使用したいと思い，3Dアトリエ(体験版)を使って作ってみたのですが，バイナリ形式ファイルのせいでしょうか，テクスチャできるモデルが作れません。知っている方，教えてくれませんか？eメールアドレスは，kanaisun@d7.dion.ne.jpです。

CHECKER FLAG

つぐ美：DirectAnimationの面白い機能をたくさん盛り込んだ作品ですね。

編：ただ，DirectXのバージョンによっては，Embossなどのトランスフォームが削除されているため，エラー表示されることがあります。このときはリストから，エラー表示されたトランスフォームの名前を削除すれば正しく実行されます。念のため編集部で修正したファイル(picture2.htm)も一緒に収録してあります。

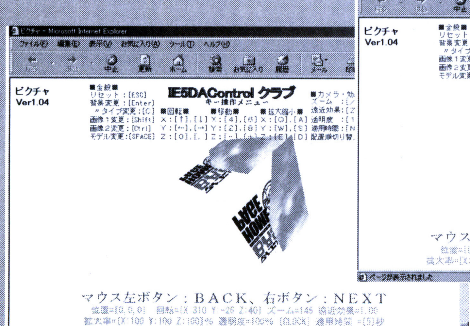
Dr.D：多くの機能を盛り込むのもよいが，少しシェイプアップして誌面にリストが載せられるくらいの容量のほうが，DirectAnimationらしいとワシは思うゾ。ただ，数多くの機能をもっておるので，参考になるリストではある。また3Dモデルについては，詳しい状況がわからんが，Xファイル自体にテクスチャ情報があったりするとテクスチャが貼れないことがあるゾ。



《写真1》3Dモデルのテクスチャに対して…



《写真2》様々なトランスフォームを適用できる



《写真3》3Dモデルや画像はお好みに変更可能

★夏盛り　暑さで動きが　スロー猛暑ン。（長野県松本市・TAC25歳）……【**影**：うまい！ベーマガ10冊！**編**：そこで，お返しの句を読むのが礼儀ですよ。**影**：この熱さの中で，私に頭を使わせるきか。**編**：たまには頭を使ったって…。**影**：スペースがなくなったー。】

Windows(C++Builder4)用



先を読んだ移動がクリアへの道

らむしふと

赤羊旅団（16歳）

ご挨拶

どうも，赤羊旅団の肉眼鏡Ｓです。この『らむしふと』は，旅団の２作目にして初のパズル・ゲームです。こういうのってアイデア勝負なんでしょうけど，ワタクシの頭が悪いのかひどく単純なものになってしまいました。こんなゲームでも，楽しんでいただければ幸いです。

ルール

(1)主人公は上下左右に動くが，一度動き始めたら羊に重なるまで止まれない。

(2)主人公と重なった羊は消える。また，ゲーム開始直後を除いて後ろには動けない。

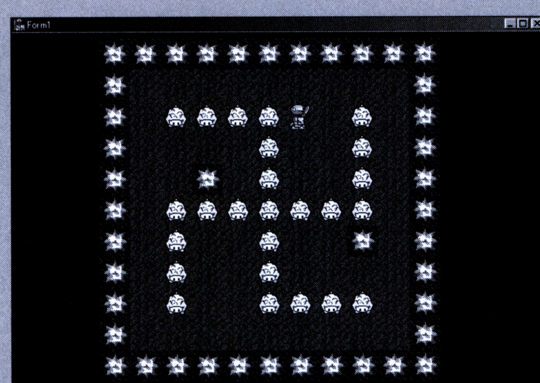
(3)画面内の羊をすべて消すとクリア！面は全8ステージ。つぐ美さんはパズルが得意らしいですが，結構気合いを入れて面を作ったので，なかなか楽しめるのではないでしょうか（8面しかないけど）？

操作の説明

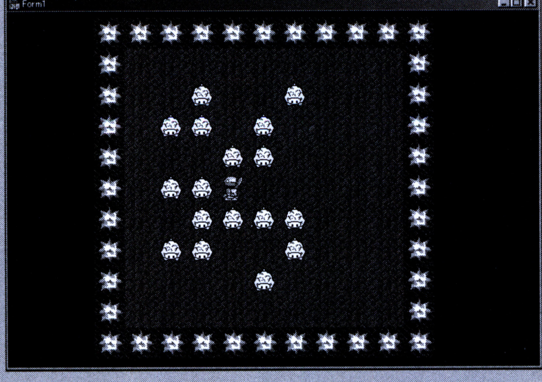
wキー：上へ移動
sキー：下へ移動
aキー：左へ移動



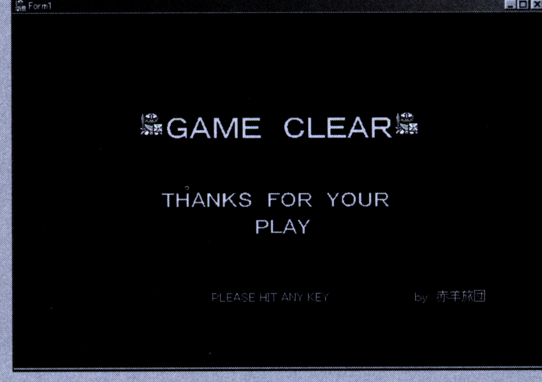
《写真１》画面上にいる羊の場所へ移動していく



《写真２》中盤には特徴的なステージも登場



《写真３》これが最終ステージ。よく考えれば道が見える？



《写真４》我らがつぐ美ちゃんは全クリ。もの足りないというお言葉も？

★もし金田一少年がプログラマーだったら，「バグはすべてとれた」と言うにちがいない。（愛媛県東宇和郡・**毛利祐允25歳**）……
【**編**：もし，影さんが名探偵だったら，事件は何も解決されないだろう。**つぐ美**：よかったわね，ベーマガ読者のアイドル＆遊び相手で。】

CHECKER FLAG

影：ゲームシステムは単純だけど，ステージがしっかり作られているので，パズルが好きな人はやりこめるゾ。ただ，その分クリアするのがけっこう難しいかも？

つぐ美：クリアのヒントとしては，一筆書きをしていけばいいので，ステージ上にある羊を線で結んでいくように考えるとクリアしやすいわよ。独立した場所にある羊を中心に考えるのがコツかしら？

Dr.D：テキストファイルで作成した面データを読み込むようにしてあるため，オリジナルのステージがテキストエディタで作成できるゾ。

編：ゲームバランスについては，Programmer's Roomのページでも触れますが，個人的な見解としてはゲームの遊びやすさだと考えています。説明文の改良の項目で回数制限についてふれていますが，こういう機能の有無を考えることが，ゲームバランスを考えることになるのではないでしょうか？

dキー：右へ移動

bキー：一手前に戻す（最高８手まで可能)。

仕組み

一手戻すにはリングバッファとやら（2001年6月号の「Catlikeのもっと身近にプログラミング」を参考）を使ってみました。どうやら，うまく動いたようなのでホッとしています。

改良について

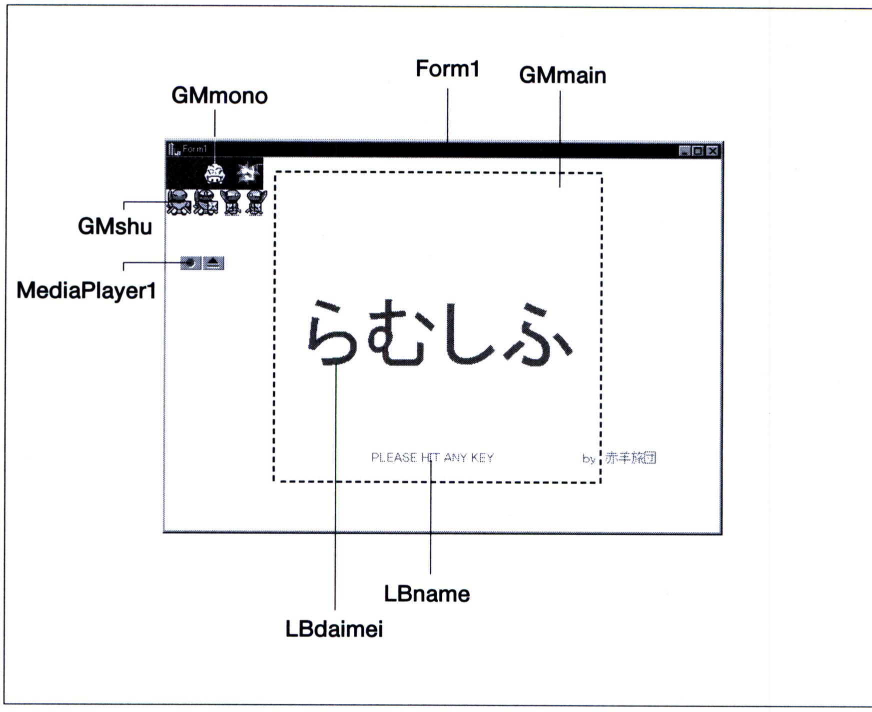
角を持った羊とか，スイッチとか，こちらが動くごとに移動する針とか……。あ，クリアタイムやミスの回数は，カウントしたほうが良かったかな？今のままだと，何度失敗してもＯＫなので。んん，でも回数制限あると，うっとおしいかなぁ。どうなんだろう（未熟)。

その他

前回アクション・ゲーム『フリスクフリッター』を送らせていただいたのですが，「もう一歩」でしたか…。ううむ，いきなり超大作を作ったのが無謀だったかな。

それにしても，ゲームバランスってなんですか？なんか，抽象的でよくわからないんです。では，また投稿します。

☆　☆　☆



フォームの配置図

《第１表》プロパティ表

フォームの設定	
プロパティ	
Name	Form1
ClientHeight	459
ClientWidth	688
Color	clBlack
Height	486
Left	370
Top	162
Width	696
イベント	
OnCreate	FormCreate
OnKeyPress	FormKeyPress
イメージの設定	
Name	GMmain
Height	440
Left	112
Top	8
Width	440
Name	GMmono
Autosize	true
Height	40
Left	0
Picture	ビットマップファイル
Top	0
Width	120
Name	GMshu
Autosize	true
Height	64
Left	0
Picture	ビットマップファイル
Top	40
Width	128
ラベルの設定	
Name	LBdaimei
Caption	らむしふと
Color	clBlack
Height	97
Left	168
ParentFont	false
Top	168
Width	346
Name	LBname
Caption	HIT ANY KEY　　by赤羊旅団
Color	clBlack
Height	16
Left	256
ParentFont	false
Top	360
Width	355
メディアプレイヤーの設定	
Name	MediaPlayer1
Height	20
Left	16
Top	120
Visible	false
VisibleButton	[btRecord,btEject]
Width	57

《第２表》関数・変数表

関数	
yomitori()	mmen(今何面なのか記憶）から面のデータを読み取る
kaku()	主人公と背景の絵を書く
idoukaku()	移動中の絵を描きます。kaku()との違いは，背景全部を書き直さないで主人公周辺だけを書き直すこと
sousa()	キーが押されたら，それにあわせて主人公を移動させる
hantei()	１マス移動したら，そこがどんなマスか調べる
end()	しょぼいエンディングを実行します
Woto(name)	name.wavを鳴らします
FormKeyPress()	押されたキーを取得します
clear()	羊を全部倒したときの処理です
bad()	針にあたったときの処理です
変数	
sj	今，キーを待っているのか判定しているのか，リングバッファに書き込んでいるのか，というのを記憶
WAHTKEY	どのキーが押されたのか記憶
men	今何面か記憶
shu.kill	倒した羊の数
.X	主人公のX座標
.Y	主人公のY座標
.it	主人公の「移動，縦」(Y移動量)
.iy	主人公の「移動，横」(X移動量)
.anime	主人公の向き
back[0-9].word	押したキーを記憶
.mx	そのときの主人公のX座標
.my	そのときの主人公のY座標
b	リングバッファ（back[]）のどこまで書き込んだか記憶
bcu	何回「一手戻す」が実行されたか記憶（移動と共に減る）
map[ ][ ]	マップのデータ
tkMX	敵（羊）の出現数

★あくせすROOMが2ページあるとモーレツにうれしくなるのは私だけじゃない，ハズ。（愛知県知多郡・**四谷一獲26歳**）……【**編**：編集部でも2ページにしたいのですが，投稿量や，連載記事の兼ね合いで，やむを得ず1Pなんです。もうしばらくお待ちください。】

《リスト１》Project1.cppのプログラムリスト

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop
USERES("Project1.res");
USEFORM("Unit1.cpp", Form1);
//---------------------------------------------------------------------------
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
        try
        {
                Application->Initialize();
                Application->CreateForm(__classid(TForm1), &Form1);
                Application->Run();
        }
        catch (Exception &exception)
        {
                Application->ShowException(&exception);
        }
        return 0;
}
//---------------------------------------------------------------------------
```

《リスト２》Unit1.cppのプログラムリスト

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop
#include "math.h"
#include "Unit1.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TForm1::AppIdle(TObject *Sender,bool &Done)
{
 switch(sj)
 {case 0:if(WHATKEY!=' '){                          //最初、最後のキー待ち
           sj=1;LBname->Visible=false;men=0;
                LBdaimei->Visible=false;
         }
         break;
  case 1:yomitori();shu.kill=0;break; //初期化
  case 2:kaku();WHATKEY=' ';sj++;break;//絵を書く
  case 3:sousa();break;               //キー待ち
  case 4:back[b].word=WHATKEY;back[b].mx=shu.mx;back[b].my=shu.my;sj++;
         b++;if(b==10){b=0;}         //一つ戻す処理
         if(bcu!=0){bcu--;}break;
  case 15:hantei();break;            //移動後の処理
  case 16:end();
  break;
  default:shu.X+=shu.iy;shu.Y+=shu.it;sj++;
          idoukaku();                //移動中の処理
 }
 Done=false;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::yomitori()          //マップを読み取る
{
  sj=2;shu.it=shu.iy=shu.kill=0;back[0].word=back[1].word=back[2].word=0;
  char buff[35000];AnsiString mmen=men;
  int hFile = FileOpen("m" + mmen + ".txt",fmOpenRead); //ファイル名指定して読み取り
    if(hFile!=-1){
       int len = FileRead(hFile,buff,sizeof(buff));
       FileClose(hFile);
    }
    for(int kk=0;kk<11;kk++){                         //map[X][Y]へ代入
        for(int k=0;k<11;k++){
         map[k][kk]=buff[kk*13+k];
    }   }
    tkMX=buff[143]-48;                                //倒すべき羊総数
    shu.mx=buff[145]-48;shu.X=shu.mx*40+4;            //主人公最初の位置へ
    shu.my=buff[146]-48;shu.Y=shu.my*40+4;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::kaku()
{
  for(int i=0;i<11;i++){
  for(int ii=0;ii<11;ii++){int m=map[ii][i]-48;    //背景描く
  BitBlt(GMmain->Canvas->Handle,ii*40,i*40,40,40,GMmono->Canvas->Handle,40*m,0,cmSrcCopy);
  } }                                                //主人公描く(↓)
    BitBlt(GMmain->Canvas->Handle,shu.X,shu.Y,32,32,GMshu->Canvas->Handle,shu.anime*32,32,c
mSrcErase);
    BitBlt(GMmain->Canvas->Handle,shu.X,shu.Y,32,32,GMshu->Canvas->Handle,shu.anime*32,0,cm
SrcInvert);
   GMmain->Refresh();
}
//---------------------------------------------------------------------------
void __fastcall TForm1::idoukaku() //移動中の絵を描く
{
  for(int k=0,kx=0,ky=0;k<2;k++,kx+=shu.iy/4,ky+=shu.it/4) //主人公のいる所の背景を描く
  {int m=map[shu.mx+kx][shu.my+ky]-48;
   BitBlt(GMmain->Canvas->Handle,(shu.mx+kx)*40,(shu.my+ky)*40,40,40,GMmono->Canvas->Handle
,m*40,0,cmSrcCopy);
  }
  BitBlt(GMmain->Canvas->Handle,shu.X,shu.Y,32,32,GMshu->Canvas->Handle,shu.anime*32,32,cmS
rcErase);
  BitBlt(GMmain->Canvas->Handle,shu.X,shu.Y,32,32,GMshu->Canvas->Handle,shu.anime*32,0,cmSr
cInvert);
  GMmain->Refresh();                                   //主人公を描いて再描写
}
//---------------------------------------------------------------------------
void __fastcall TForm1::sousa()
{ int db=b;db--;if(db<0){db=9;}
  if(WHATKEY!=' '){
     switch(WHATKEY)   //押されたキー確認
      {
        case 'w':if(back[db].word!='s'){shu.it=-4;shu.anime=0;sj++;}break;//上
        case 's':if(back[db].word!='w'){shu.it=4; shu.anime=1;sj++;}break;//下
        case 'a':if(back[db].word!='d'){shu.iy=-4;shu.anime=2;sj++;}break;//左
        case 'd':if(back[db].word!='a'){shu.iy=4; shu.anime=3;sj++;}break;//右
      }

  }
  if(WHATKEY=='b' && bcu<8 && shu.kill>0){               //'b'が押された。
     sj=2;b--;bcu++;shu.kill--;map[shu.mx][shu.my]=49; //現在地に羊を置いて、倒した数-1
     if(b<0){b=9;}
     shu.mx=back[b].mx;shu.my=back[b].my;
     shu.X=shu.mx*40+4;shu.Y=shu.my*40+4; //前の場所へ移動
     db--;
     if(db<0){db=9;}
     switch(back[db].word){            //体の向き調整
        case 'w':shu.anime=0;break;
        case 's':shu.anime=1;break;
        case 'a':shu.anime=2;break;
        case 'd':shu.anime=3;break;
     }
```

★Visual C++のプログラミング講座を開いてください。Ver5だとなお良し。（埼玉県北葛飾郡・**鹿谷浩一18歳**）……【**編**：読者の意見第一のベーマガです。C言語講座開始に向け，内容や連載時期など，現在打ち合わせ中です。みなさんの意見もお待ちします。】

★（三國志Ⅷネタ）「今一番人気を集めているのは、ともこぎるさんか」「たしかに」「だが、南斗ライバ氏もその名をとどろかせてきた」「畢洋平もなかなかがんばっておるな」（おいおい）。（群馬県高崎市・**ぼりごんはーと22歳**）……**【影：おい、私を忘れちゃ困るぞ。】**

```
 }
 }
//------------------------------------------------------------------------
void __fastcall TForm1::hantei()
{
 shu.mx+=shu.iy/4;shu.my+=shu.it/4;
 switch(map[shu.mx][shu.my]-48)        //主人公の現在地が……
 {
  case 0:sj=5;break;                          //何もなし。次のマスへ。
  case 1:back[b].word=WHATKEY;                //    羊     倒してキーを待つ
         shu.kill++;shu.it=shu.iy=0;map[shu.mx][shu.my]=48;Woto("kill");
         if(tkMX==shu.kill){clear();}else{sj=2;}
  break;                                      //    針      BAD関数へ。
  case 2:bad();break;
} }
//------------------------------------------------------------------------
void __fastcall TForm1::FormKeyPress(TObject *Sender, char &Key)
{
 WHATKEY=Key;                  //WHATKEYに押されたキーを代入
}
//------------------------------------------------------------------------
void __fastcall TForm1::FormCreate(TObject *Sender)
{
 bcu=8;WHATKEY=' ';
 Application->OnIdle = AppIdle;
}
//------------------------------------------------------------------------
void __fastcall TForm1::Woto(String name)
{
 sndPlaySound(name.c_str(),SND_ASYNC|SND_NODEFAULT);//音を鳴らす
}
//------------------------------------------------------------------------
void __fastcall TForm1::clear()
{
 WHATKEY=' ';
 Woto("clear");men++;sj=1;Sleep(2700);  //音を鳴らして次の面へ
 if(men==8){sj=16;}                      //全面クリアならエンディングへ
}
//------------------------------------------------------------------------
void __fastcall TForm1::bad()
{
 Woto("bad");sj=1;Sleep(200);back[b-1].word=' '; //音を鳴らしてやりなおし
}
//------------------------------------------------------------------------
void __fastcall TForm1::end()  //しょぼいエンディング
{
 GMmain->Canvas->Brush->Color=clBlack;
 GMmain->Canvas->Rectangle(0,0,440,440);
 GMmain->Canvas->Font->Color=clRed;
 GMmain->Canvas->Font->Size=30;
 GMmain->Canvas->TextOutA(85,120,"GAME  CLEAR");
for(int k=0;k<2;k++){
 BitBlt(GMmain->Canvas->Handle,50+k*330,120,32,32,GMshu->Canvas->Handle,32,32,cmSrcErase);
 BitBlt(GMmain->Canvas->Handle,50+k*330,120,32,32,GMshu->Canvas->Handle,32,0,cmSrcInvert);
}GMmain->Refresh();
 Sleep(1000);
 GMmain->Canvas->Font->Size=20;
 GMmain->Canvas->TextOutA(80,220,"THANKS  FOR  YOUR");
 GMmain->Canvas->TextOutA(200,255,"PLAY");
 GMmain->Canvas->Font->Size=12;
 GMmain->Refresh();
 Sleep(1000);
 LBname->Visible=true;
 sj=0;
}
```

《リスト3》m0.txtのプログラム・リスト

```
22222222222
20000000002
20010000002
20110000002
20010010002
20011010002
20010010002
20000000002
20010010002
20000000002
22222222222
< 28
```

《リスト4》m1.txtのプログラム・リスト

```
22222222222
20000000002
20010011002
20011001002
20100100002
20110001002
20011011002
20000000002
20000000002
20000000002
22222222222
? 44
```

《リスト5》m2.txtのプログラム・リスト

```
22222222222
20000000002
20101000002
20101000002
20111110002
20120000002
20111110002
20101000002
20101000002
20000000002
22222222222
C 45
```

《リスト6》m3.txtのプログラム・リスト

```
22222222222
20000011002
20100001002
20100010002
20000111102
20000100002
20000111002
20100010002
20100010102
20000000002
22222222222
C 83
```

《リスト7》m4.txtのプログラム・リスト

```
22222222222
20000000002
20000000002
20110101002
20010101002
20110111102
20010101002
20110101002
20000000002
20000000002
22222222222
D 45
```

《リスト8》m5.txtのプログラム・リスト

```
22222222222
20000000002
20000000002
20010100002
20111110002
20010100002
20111110002
20000100002
20000000002
20000000002
22222222222
? 37
```

《リスト9》m6.txtのプログラム・リスト

```
22222222222
20000000002
20111100102
20000100102
20020100102
20111111102
20100100202
20100100002
20100111102
20000000002
22222222222
I 62
```

《リスト10》m7.txtのプログラム・リスト

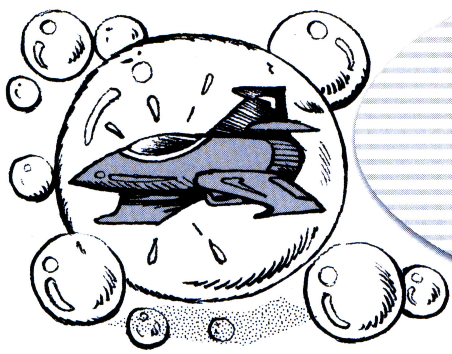
```
22222222222
20000000002
20010010002
20110100002
20001100002
20110000002
20011110002
20110010002
20000100002
20000000002
22222222222
A 45
```

評…久々の投稿ありがとう。部活もがんばっ！（編）

by 旭野夕海



Windows(HSP V2.5)用



反動で動くキャラクタが特徴

シャボン狩り（シャボンが狩り）

畑の木（28歳）

はじめに

HSPには円を描くコマンドがないので，ちょっと自作してみました。ゲームのほうは，おまけです。…の割には長くなったけど。

モジュールについて

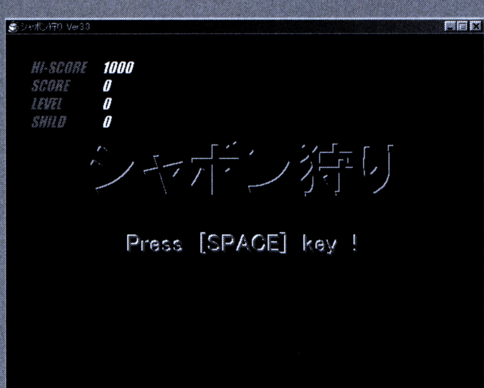
円を描くモジュールです。必要なら，#includeで結合させて使ってください。命令の名前，ならびに引数は，circle p1,p2,p3と宣言しています。

p1(パラメータ1)で円の中心のX座標，p2はY座標，p3は半径を指定します。色はcolorかpalcolorで設定されたものになります。ただし，実際に描かれる円の直径は，半径を倍した数に1加えた値になります（つまり，半径を50にした場合，50(半径)+1(中心点)＋50(半径)＝101(直径)になる）ので注意してください。

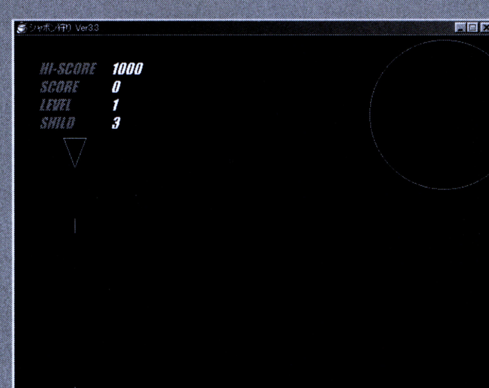
また，このアルゴリズムは有名なものらしいので，興味のある方は図書館でその手の本を探してみてください（私にもよくわかっていないので説明できない）。

おまけのゲーム

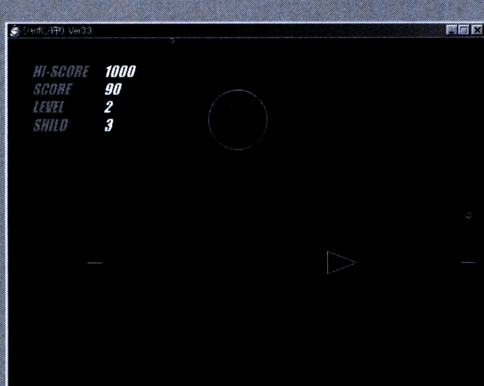
自機を操って，敵の円（シャボン）を破壊していくゲームです。自機の操作に特徴があって，カーソル・キーの上下左右で向きを変え，スペース・キーで弾を撃つことで，その反動を利用し移動します。シャボンは色によって，耐久力，スピード，動きかた，得点が変わってきま



《写真1》シャボン玉風の敵を倒していく



《写真2》発射する弾の反動で移動する



《写真3》シャボンはダメージやレベルで変化していく



《写真4》高得点を目指してプレイしよう

★愛するコンピュゥラァ，君のために，今日も僕はモチを撃つ。（愛知県名古屋市・**ズオウ・リアキ29歳**）……【**Dr.D**：最近はワシもなかなかついていけないぞ。**影**：いまにはじまったことじゃないでしょ，Dr.。**Dr.D**：こんな影君のためにワシは今日もハリを打つ。】

CHECKER FLAG

影：基本的なゲームに見えるけど，キャラクターの移動方法が弾を発射した反動で動かすというユニークな方法なのが特徴だ。

つぐ美：ただ，全体的に単調な進行なのが気になりました。あと，たまに弾が当たらないことがありますね。当たり判定ルーチンを見直してみてはどうかしら？

編：命令モジュールだけでなく，それを用いたサンプル・ゲームも一緒に投稿してくれているのがウレシイですね。モジュール関連は，まずは作成した本人自らのサンプルがほしいですからね。

Dr.D：円描画という基本的な命令じゃが，使われているアルゴリズムは興味深いものじゃ。わかりやすい流れでしかもしっかり円が描画される。点を打つ命令があれば，おそらくどの言語でも実現可能じゃろう。

す。シャボンには，ほかに大きさによる違いがあり，一番小さいしゃぼんを全滅させると，レベルアップして新しいシャボンが現れます。レベル5くらいまでは，それほど難しくありませんが，レベル10以上に行くのはかなり大変です。

自機は，敵にぶつかるとシールドが一つ減り，シールド０で敵にぶつかると，ゲーム・オーバーになります。自機ならびにシャボンも，点滅している間は無敵です。

タイトル画面でも，ゲーム中でもESCキーでプログラム終了です。

注意点

ゲームのプログラムと同じフォルダに，円を描くモジュール(circle.as)を保存しておいてください。ビットマップファイルは必要ありません。それから，このモジュールは非常に遅いので注意してください。このゲームでは，いったん仮想画面に描いて利用しています（ループ毎，モジュールで円を描いていたら，すごく遅くなったので…)。

改造のポイント

音をつけると，格段に面白くなると思います。ほかに，シャボンの動きのバリエーションを増やしたり，速度に余裕があれば，１画面中に登場する敵の数を増やしたり。

あと書き

それにしても長いプログラムになりました。これでも，削ったんですけど。また何かできたら送ります。

★参考にしたもの

・『ＳＡＳＡ』アスキー

《第１表》敵データ表

緑	動きが遅く，規則どおりにバウンドしているだけ。最弱
青	動きが遅く，壁にぶつかったとき，自機の側にバウンドする。 操作になれていれば、それほど恐くない
黄	動きが速く，規則どおりにバウンドしている。 距離をとって攻撃するのが倒すコツ
赤	動きが速く，壁にぶつかると，自機の側にバウンドする。最強

《第２表》変数表

p_x	プレイヤーのX座標
p_y	プレイヤーのY座標
p_r	自機の向き
p_c	自機の無敵時間用
shot_n	現在の画面上のショットの数
shot_t	ショットを撃ってからの時間計測用
（以下，n,n1,n2は配列番号)	
shot_s.n	ショットの向き（-1なら画面上にない)
shot_x.n	ショットのX座標
shot_y.n	ショットのY座標
e_k.n	しゃぼん（敵）の色
e_s.n	しゃぼんの半径（100，70，40のどれか)
e_sn.n	しゃぼんの大きさの番号（0,1,2のどれか)
e_h.n	しゃぼんの耐久力
e_c.n	しゃぼんの無敵時間用
e_x.n	しゃぼんのX座標
e_y.n	しゃぼんのY座標
e_dx.n	しゃぼんのX方向の速度
e_dy.n	しゃぼんのY方向の速度
e_dat.n1.n2	レベル毎のしゃぼんデータ

《リスト１》Circle.asのプログラム・リスト

```
;///////////////////////////////////////////////////////////////////////////
;       円を描くためのモジュール For HSP
;///////////////////////////////////////////////////////////////////////////

#module "_circle"

;       circle p1, p2, p3
;       p1: 円の中心のｘ座標
;       p2: 円の中心のｙ座標
;       p3: 円の半径
;
;点(p1,p2)を中心とした、半径p3の円を描く。
;絵画色はcolor命令、またはpalcolor命令で設定されたものになる。
;ただし、この場合の半径は、中心の点を含まない。
;（例えば、半径を100とした時、直径は（半径）100 +（中心点）1 +（半径）100 = 201 になる）

#deffunc circle int, int, int

        mref center_x, 0                                    ;ｘ座標代入
        mref center_y, 1                                    ;ｙ座標代入
        mref r, 2                                           ;半径代入

        y = -r: c = 1 - r
        px = 0: py = y: gosub *circle_pset
        repeat r * 1000 / 1414 + 1                          ;円を８分の１描く
                x = cnt
                if c < 0 {
                        px = x + 1: py = y
                        gosub *circle_pset
                        c += 2 * x + 3
                }
                else {
                        px = x + 1: py = y + 1
                        gosub *circle_pset
                        c += 2 * (y + x) + 5
```

★ええええっえ（8月号95ページ）？！Dr.Jって，つぐ美さん？そんなわけはないと思うが。(長野市・**HoHo13歳**)……【**つぐ美**：あのイラストね。正体がわかってしまったようですよ，Dr.。**Dr.D**：ワシよりDr.Jネタのほうが多いように思うのじゃが，気のせいかのう。】

```
                y++
            }
        loop
        return

*circle_pset    ;８個の対称点をとる
        pset  px + center_x,  py + center_y: pset  px + center_x, -py + center_y
        pset -px + center_x,  py + center_y: pset -px + center_x, -py + center_y
        pset  py + center_x,  px + center_y: pset  py + center_x, -px + center_y
        pset -py + center_x,  px + center_y: pset -py + center_x, -px + center_y
        return
#global
```

《リスト２》シャボン狩りのプログラム・リスト

```
;///////////////////////////////////////////////////////////////////////////////////////////
//////
;       シャボン狩り ver3.3  for HSP 2.5
;       (c) 畑の木
;///////////////////////////////////////////////////////////////////////////////////////////
//////

#include "circle.as"                    ;---------- "circle.as"を結合させる
        title "シャボン狩り Ver3.3"     ;---------- タイトルバーの表示
        randomize                       ;---------- 乱数の初期化
        dim shot_s, 4                   ;---------- ショットの状態を保存
        dim shot_x, 4: dim shot_y, 4
        dim e_k, 12: dim e_s, 12: dim e_sn 12: dim e_h, 12: dim e_c, 12 ;------ 敵データ
        dim e_x, 12: dim e_y, 12: dim e_dx, 12: dim e_dy, 12
        dim e_dat, 3, 10                ;---------- レベル別の敵の種類データ用
        e_dat     = 0, 0, 0, 0, 0, 1, 0, 1, 4, 0, 5, 5, 1, 1, 2
        e_dat.0.5 = 2, 1, 6, 2, 6, 6, 0, 5, 6, 1, 6, 7, 3, 7, 7
        hi_score = 1000                 ;---------- 初期のハイスコア
        buffer 1, 800, 840, 1           ;---------- ウィンドウID１用意
        cls 4   ;---------- 黒く塗りつぶす
        color 255, 255, 255
        line 20, 0, 35, 39: line 35, 39, 5, 39: line 5, 39, 20, 0      ;----- 自機を描く
        line 40, 20, 79, 5: line 79, 5, 79, 35: line 79, 35, 40, 20
        line 85, 0, 115, 0: line 115, 0, 100, 39: line 100, 39, 85, 0
        line 120, 5, 159, 20: line 159, 20, 120, 35: line 120, 35, 120, 5
        repeat 3
                n = cnt         ;---------- 円（敵）を描く
                color 64, 255, 0: circle 100, n * 200 + 140, 100 - (30 * n) - 1
                color 0, 128, 255: circle 300, n * 200 + 140, 100 - (30 * n) - 1
                color 255, 255, 0: circle 500, n * 200 + 140, 100 - (30 * n) - 1
                color 255, 32, 0: circle 700, n * 200 + 140, 100 - (30 * n) - 1
        loop
        screen 0, 640, 480, 1, dispx / 2 - 320, dispy / 2 - 240 ;---------- スクリーン設定
        gsel 0, 1       ;---------- ウィンドウ0をアクティブに（exeファイル作成用）
        gmode 2         ;---------- 画面コピーモード設定
*title_draw                             ;---------- タイトル画面を描く
        redraw 0                        ;---------- 仮想画面に描く
        color 0, 0, 0: boxf 0,0,640,480 ;---------- 画面を黒く塗りつぶす
        font "MS 明朝", 80, 0                           ;---------- フォント
        color 64, 32, 255: pos 108, 140: mes "シャボン狩り"     ;---------- 青で文字を書く
        color 0, 0, 0: pos 110, 142: mes "シャボン狩り"         ;---------- 黒で文字を書く
        font "MS 明朝", 32, 0                           ;---------- フォント
        color 255, 255, 255: pos 160, 260: mes "Press  [SPACE]  key  !" ;----- 白の文字
        color 128, 128, 128: pos 162, 262: mes "Press  [SPACE]  key  !" ;----- 灰色の文字
        level = 0: shild = 0: score = 0: gosub *write_score     ;---------- スコア表示
        redraw 1                        ;---------- 画面の更新
        repeat
                wait 1                  ;---------- ウェイト
                stick key               ;---------- キー入力待ち
                if key&112: break       ;---------- スペースキー（スタート）
                if key=128: end         ;---------- ESCキー（終了）
        loop
*game   ;---------- ゲーム部分
        p_x = 320: p_y = 240: p_r = 0: p_c = 0  ;---------- 自機の初期
        shot_n = 0: shot_t = 0                          ;---------- ショットの初期設定
        repeat 4: shot_s.cnt = -1: e_h.cnt = 0: loop    ;----------
        shild = 3: score = 0: first_wait_flag = 1       ;---------- 初期変数設定
        gosub *level_up                                 ;---------- レベルを１にする
        repeat
                redraw 0                                ;------- 仮想画面へ
                        color 0, 0, 0: boxf 0, 0, 640, 480      ;------- 全面黒くする
                        gosub *write_score              ;------- スコア他を表示へ
                        gosub *key_check                ;------- キーチェックへ
                        gosub *shot                     ;------- ショット処理へ
                        gosub *enemy                    ;------- 敵の処理へ
```


★「Dr.D」と「Dr.J」，2人あわせて「Dr.DJ」…って，「ヤン坊」「マー坊」かいっ？（茨城県古河市・**年齢不相応41歳**）……【**Dr.D**：まあ，従兄弟同士じゃから，2人で売り出すのも悪くないかな。**つぐ美**：そのわりには，顔が引きつってますよ，Dr.。**Dr.D**：むむ。】

```
			gosub *hantei
		redraw 1					;----- 画面の更新
		if first_wait_flag: first_wait_flag = 0: wait 50;----- 一番最初のウェイト
		await 1						;----- 通常ならこのウェイト
		if shild < 0: break			;---- シールドが0未満ならループを抜ける
	loop
	gosub *gameover		;---------- ゲームオーバーへ
	goto *title_draw	;---------- タイトルに戻る
*key_check	;---------- キーチェック ＆ 自機の表示
	stick key, 16		;---------- キー入力
	if key & 1: p_r = 1	;---------- 左が押されたか？
	if key & 2: p_r = 0	;---------- 上が押されたか？
	if key & 4: p_r = 3	;---------- 右が押されたか？
	if key & 8: p_r = 2	;---------- 下が押されたか？
	if key = 128: end	;---------- ESCが押されたら、終了

	if (p_c ¥ 2): return	;---------- 自機の点滅処理
	pos p_x - 20, p_y - 20: gcopy 1, p_r * 40, 0, 40, 40	;---------- 自機を描く
	return
*shot		;---------- ショットの処理
	if first_wait_flag: return	;----- 最初の待ち時間、ショットを表示させないため
	if (key & 16) * (shot_n < 4) * (shot_t = 0) {	;---------- ショット発射か？
		shot_t = 30: shot_n += 1	;---------- ショットの数を増す
		repeat 4
			n = cnt
			if shot_s.n < 0 {;----- 何番目の配列にショットのデータを登録するか
				shot_s.n = p_r	;------- ショットの向き＝現在の自機の向き
				shot_x.n = 10 * ((p_r = 3) - (p_r = 1)) + p_x	;位置設定
				shot_y.n = 10 * ((p_r = 2) - (p_r = 0)) + p_y
				break
			}
		loop
	}
	if shot_t > 0 { ;----------- 自機にかかるショットの衝撃
		shot_t--		;---------- ショット発射からのカウンター
		p_x += 2 * ((p_r == 1) - (p_r == 3))	;------- 向きより自機の位置を更新
		p_y += 2 * ((p_r == 0) - (p_r == 2))	;----------
		p_x -= ((p_x < 0) * p_x) + ((p_x > 640) * (p_x - 640))	;------- 壁の処理
		p_y -= ((p_y < 0) * p_y) + ((p_y > 480) * (p_y - 480))
	}
	repeat 4
		n = cnt
		if shot_s.n >= 0 {	;---------- 現在、表示されているショットか？
			shot_x.n += 4 * ((shot_s.n == 3) - (shot_s.n == 1))	;位置の更新
			shot_y.n += 4 * ((shot_s.n == 2) - (shot_s.n == 0))
			color 255, 255, 255
			x = 20 * ((shot_s.n = 3) - (shot_s.n = 1)) + shot_x.n
			y = 20 * ((shot_s.n = 2) - (shot_s.n = 0)) + shot_y.n
			line shot_x.n, shot_y.n, x, y	;---------- ショットの表示
			if (shot_x.n<=0) + (shot_x.n>=640): shot_s.n=-1: shot_n--;消す？
			if (shot_y.n<=0) + (shot_y.n>=480): shot_s.n=-1: shot_n--;
		}
	loop
	return
*write_score	;---------- スコアを書く
	if score > hi_score: hi_score = score	;---------- スコアとハイスコアの比較
	font "impact", 20, 2: color 255, 0, 128: pos 32, 32	;---------- フォント他
	mes "HI-SCORE¥nSCORE¥nLEVEL¥nSHILD"	;---------- 文字表示
	color 255, 255, 255: pos 128, 32	;---------- 数値を書く位置
	mes "" + hi_score + "¥n" + score + "¥n" + level + "¥n" + shild	;------- 数値表示
	return

*level_up	;---------- レベルアップ
	level++:if (level ¥ 3) = 0: shild++	;- レベルアップ、シールドを増加させるか？
	s = 0: n = 0: e_n = 1: gosub *new_enemy:return	;---------- 新しい敵の設定へ

*new_enemy	;---------- 新しい敵の設定
	e_sn.n = s: e_s.n = 100 - (30 * e_sn.n) ;---------- 敵のサイズの設定
	j = level - 1: if j > 9: j = 8 + (j ¥ 2)	;---------- レベルが一定以上なら
	e_k.n = e_dat.s.j				;---------- 敵の色
	if e_k.n > 3: bunretu = 1: e_k.n -= 4: else: bunretu = 0;---------- 分裂するか？
	e_h.n = 6 + (e_k.n * 2)			;---------- 敵の耐久力
	e_dx.n = 1 + (e_k.n > 1): e_dy.n = 1 + (e_k.n > 1)	;---------- スピード設定
	rnd k, 4
	if k ¥ 2: e_dx.n = -e_dx.n		;---------- 乱数から、敵の動く向きを定める
	if k > 1: e_dy.n = -e_dy.n		;----------
	e_c.n = 250				;---------- 新しい敵の無敵時間用

	if s = 0 {	;---------- 一番大きい敵か？
		repeat
			rnd e_x.n, 640: rnd e_y.n, 480	;---------- 新しい敵の位置候補
```

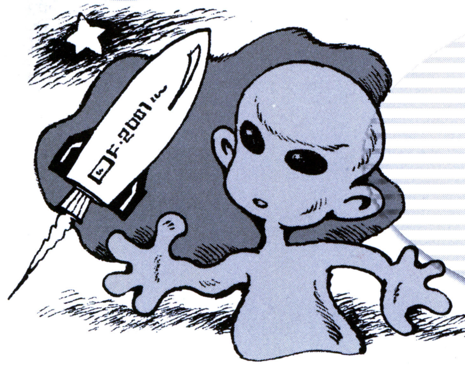

★情報処理演習の授業でC言語を習った。これを機会にCを勉強しよう！（大阪府河内長野市・**よっち18歳**）……**【編**：C言語の話題がこのところ多いように思いますが…。**Dr.D**：うむ。C言語講座の要望も高いしのう。いま，実現に向けて調整中じゃぞ。**】**

```
                dx = (e_x.n - p_x) * (e_x.n - p_x)      ;----- プレイヤーとの距離
                dy = (e_y.n - p_y) * (e_y.n - p_y)      ;-----
                r = 300
                if (dx * dx) + (dy * dy) < (r * r): break       ;充分、自機から離す
            loop
        }
        return
*enemy          ;---------- 敵の動きの制御
        repeat 4        ;---------- 敵は最大4個まで存在する
            n = cnt
            if e_h.n > 0 {  ;---------- 敵の耐久力がある（=  敵が画面上に存在する）
                if ((e_x.n < 0) * (e_dx.n < 0)) + ((e_x.n > 640) * (e_dx.n > 0)) {
                    e_dx.n = -e_dx.n: if e_k.n ¥ 2: gosub *oikake
                }
                if ((e_y.n < 0) * (e_dy.n < 0)) + ((e_y.n > 480) * (e_dy.n > 0)) {
                    e_dy.n = -e_dy.n: if e_k.n ¥ 2: gosub *oikake
                }
                e_x.n += e_dx.n: e_y.n += e_dy.n        ;---------- 敵の位置更新
                e_c.n -= (e_c.n > 0): if (e_c.n ¥ 2): continue  ;----- 敵の点滅用
                pos e_x.n - e_s.n, e_y.n - e_s.n        ;---------- 敵の表示位置
                x = e_k.n * 200 + 100 - e_s.n: y = e_sn.n * 200 + 140 - e_s.n
                gcopy 1, x, y, 2 * e_s.n, 2 * e_s.n     ;---------- 敵表示
            }
        loop
        return
*oikake ;---------- 簡易追いかけルーチン
        sign_x = 1 - (2 * (p_x < e_x.n)): e_dx.n = sign_x * (1 + (e_k.n > 1))
        sign_y = 1 - (2 * (p_y < e_y.n)): e_dy.n = sign_y * (1 + (e_k.n > 1))
        return
*hantei ;---------- ショットと敵  &  自機と敵の衝突判定
        p_c -= (p_c > 0)        ;---------- 自機の無敵時間用カウンター
        repeat 4
            n1 = cnt: if (e_h.n1 = 0) + (e_c.n1 > 0): continue ;敵がいない、敵が無敵
            repeat 4
                n2 = cnt: if shot_s.n2 < 0: continue    ;---------- ショットがない
                dx = e_x.n1 - shot_x.n2: dy = e_y.n1 - shot_y.n2
                r = e_s.n1 + 10                         ;大きめに当たり判定を取る
                if (dx * dx) + (dy * dy) < (r * r) {    ;敵とショットが当たった？
                    e_h.n1--: shot_n--: shot_s.n2 = -1
                    rnd i, 4
                    if i > 1: e_dx.n1 = -e_dx.n1        ;---------- 動きを変える
                    if i ¥ 2: e_dy.n1 = -e_dy.n1
                    if e_h.n1 = 0: gosub *break_enemy       ;敵が弾けた？
                }
            loop
            if p_c + (e_h.n1 = 0): continue ;---------- 自機が無敵か、敵がいない
            dx = e_x.n1 - p_x: dy = e_y.n1 - p_y
            r = e_s.n1 - 10             ;---------- 小さめに当たり判定を取る
            if (dx * dx) + (dy * dy) < (r * r): shild--: p_c=250 ;自機と敵が当たった？
        loop
        return
*break_enemy    ;---------- 敵が弾ける（壊れる）
        score += 10 * (e_sn.n1 + 1) * (e_k.n1 + 1)  ;---------- スコア増
        if e_sn.n1 = 2 {                            ;---------- 一番小さな敵か？
            e_n--: if e_n = 0: gosub *level_up  ;敵の数を減らし、0になればレベルアップ
            return
        }
        n = n1: s = e_sn.n1 + 1: gosub *new_enemy       ;---------- 新しい敵の設定
        if bunretu {                                    ;---------- 分裂する？
            repeat 4
                n3 = cnt
                if e_h.n3 = 0 {         ;何番目に分裂した敵のデータを入れるか？
                    e_n++: n = n3               ;---------- 敵の数を増やす
                    e_x.n3 = e_x.n1: e_y.n3 = e_y.n1        ;敵の初期座標
                    gosub *new_enemy            ;---------- 新しい敵設定
                    break
                }
            loop
        }
        return
*gameover
        shild = 0: wait 140     ;---------- シールドを0にする（表示用）
        redraw 0                ;---------- 仮想画面にする
        color 0, 0, 0: boxf 0, 0 , 640, 480     ;---------- 画面をクリア
        gosub *write_score      ;---------- スコアの表示へ
        font "MS 明朝", 32, 0   ;---------- フォント
        color 255, 0, 32: pos 230, 220: mes "GAME OVER"
        redraw 1                ;---------- 画面の更新
        wait 300                ;---------- ウェイト
        return
```

HSP

★近況報告：「Xbox」のことを勝手に「MSX3」と呼ぶ友人ができました。（千葉市・バグニンゲン18歳）……【**影**：Xboxもいよいよだね。11月が楽しみ！**編**：楽しみって，影さんの場合，自分で買わずに他人の買ったマシンを自分のモノのように使っちゃうんですもの。】

Javaアプレット用



8方向から攻めてくるヌー星人を倒せ

ヌー星人（せいじん）

～ロケット救出編～

マギヲ(21歳)

内　容

ロケットに近づいてくる正体不明のヌー星人をひたすら倒します。

使いかたと遊びかた

テンキーの1，2，3，4，6，7，8，9キーを使ってヌー星人のいる場所に合わせて押していきます。弾に制限はありません。ただひたすら打ちます。ロケットがヌー星人に当たると終わりです。1HITにつき＋1点，ミスは－10点です。普通のヌー星人は1HITで，赤いヌー星人は2HITで倒せます。

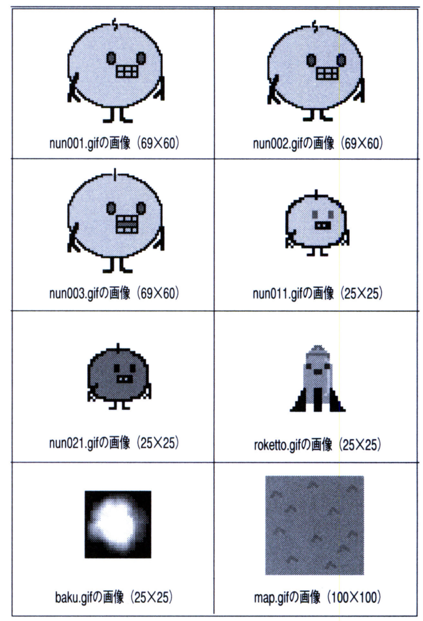
アプレットなので，ダウンロードファイルにあるNuu0100.htmlをブラウザなどで実行させます。

プログラムの仕組みと入力法

Main.classを中心にして，OpenCanvas(オープニング)，GameCanvas(ゲーム)，SccorePanel(スコア)の三つのclassをつかって構成されています。入力はテキストベースで書いてください。それをJDKのjavac.exeを使ってGameCanvas.javaをコンパイルしてください。

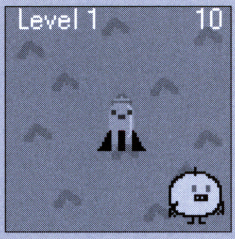
作者からのコメント

ページの片隅において楽しむゲームを作りたかったので，大きさが100×100と非常に小さいです。マジックナンバーはあまり使っていないので，変更はそう難しくないと思います。今後はiアプリでも作ってみたいな～。

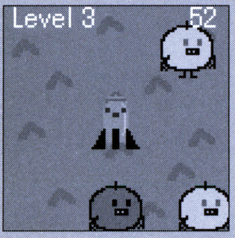




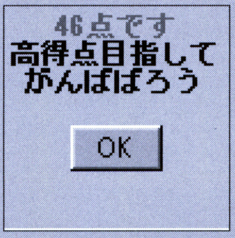
《写真１》手軽に遊べる画面サイズ



《写真２》接近してくるヌー星人を倒せ



《写真３》レベルに応じて同時出現数が増加



《写真４》減点制度もあり，高得点は出にくい

CHECKER FLAG

編：Dr.Jの講座やiアプリ関連の連載で，ベーマガにも定着しているJava。この勢いにあやかってJavaの投稿プログラムも，もっと増えていってほしいですね。

影：ゲームシステムは単純だけど，画像ファイルの読み込みやキーボードイベント処理など，ゲーム作成において必要な処理が網羅されている作品だ。

Dr.D：リストの記述面で気になったのは，キー入力の判断部分じゃ。ここはswitch文ではなく，“Key[ky-48]=1;”といったように直接配列に入れてしまったほうがシンプルじゃ。ただし，配列の範囲外に代入しないよう，直前にif文などでkyの値をチェックする必要があるゾ。また，実行用のHTMLファイルもappletタグだけでは手抜きと言われても仕方がない記述じゃゾ。

★あっつーい。今年も，どアツイ！つぐ美さ～ん，うちわであおいで～。(愛知県新城市・**竹川由美恵17歳**)……【**つぐ美：**ぱたぱた。影さん，私もあおいで。**影：**ぱたぱた。編くーん。**編：**ぱたぱた。編集長。**編集長：**ぱたぱた。Dr.。**Dr.D：**ぱたぱた。由美恵さーん。】

《リスト１》Character.javaのプログラム・リスト

```
import java.awt.*;

/* キャラクター */
public class Character extends Object {
    /* Final Opetion */
    final int KIHONPOINT = 38;              //キャラクターの位置を決めるのに使う
    final int MAX_LENGHT = 63;              //最大距離
    final int OUTLEN = 10;                  //当り判定
    final int x[] = {0,-1,0,1,-1,0,1,-1,0,1};   //移動の増減
    final int y[] = {0,1,1,1,0,0,0,-1,-1,-1};   //移動の増減
    /* Opetion */
    int Type;                               //キャラクターのタイプ
    int Where;                              //位置
    int Len;                                //距離
    int Damegi;                             //ダメージ
    /* Constructor */
    Character( int type,int whr ){
        Type    = type;
        Where   = whr;
        Damegi = --type;
        Len     = MAX_LENGHT;
    }
    public int  type()      { return Type;   }
    public int  where()     { return Where;  }
    public int  len()       { return Len;    }
    public int  damegi()    { return Damegi; }
    public int  downLen()   { return --Len;  }
    public void addDamegi(){ --Damegi;       }
    public int getX(){ return KIHONPOINT + Len * x[ Where ]; }
    public int getY(){ return KIHONPOINT + Len * y[ Where ]; }
    public boolean update(){
        downLen();
        return ( Len < OUTLEN )?true:false;
    }
    public void paint( Graphics g ,Image img,Component comp ){
        g.drawImage( img,getX(),getY(),comp );
    }
}
```

《リスト２》CharacterManager.javaのプログラム・リスト

```
import java.awt.*;

/* キャラクターマネージャ */
public class CharacterManager extends Object{
    ///////////////////
    /* Final Opetion */
    ///////////////////
    final int MAX_CHARA = 9;                            //キャラクターの数
    final int Level[] = {0,20,50,150,300,600,1100};     //LevelUpの間隔の設定
    /////////////
    /* Opetion */
    /////////////
    Component Comp;                                     //イメージ表示の為のコンポーネント
    int Teki,                                           //敵の数
        HitPoint;                                       //倒した敵の数
    Character Chctr[] = new Character[ MAX_CHARA ];    //キャラクター
    boolean Out;        //ロケットとヌー星人が当った時にｔｒｕｅを返す。
    /////////////////
    /* コンストラクタ */
    ////////////////
    public CharacterManager( Component comp ){
        Comp = comp;
        HitPoint = 0;
        Teki = LevelUp( HitPoint );
        Out = false;
        Chctr[0] = new Character( 1,5 );
        for(int i = 1;i< MAX_CHARA;i++)
            Chctr[i] = new Character( 0,1 );
    }
    ////////////////
    /* add New Teki */
    ////////////////
    public void addTeki(int point){
        int num,check;
        do{
            check = 0;
            num = Random( 1,MAX_CHARA );
            if( num != 5 ){
                for(int j=1;j<=Teki;j++)
                    if( num == Chctr[j].where() )
                        check = 1;
            }
            else check = 1;
        }while( check != 0 );
        Chctr[ point ] = new Character( Random( 2,3 ),num );
        Teki = LevelUp( ++HitPoint );
    }
    ///////////////////////////
    /* Level[]を基準にLevel Up */
    ///////////////////////////
    public int LevelUp( int poin ){
        int num = 0,ans;
        while( poin >= Level[ num ] && num <= Level.length )
            num++;
        return num;
    }
    ////////////////////////////////////
    /* minからmaxまでの整数の乱数を出す */
    ///////////////////////////////////
    public int Random( int min,int max ){
        int ans,num;
        double rn;
        if ( min >= max )
            return -1;
        rn = Math.random() * ( max - min + 1 );
        num = (int)rn + min;
        ans = ( num <= max )?num:max;
        return ans;
    }
    ////////////////////////////////////
    /* キャラクターにダメージを与える */
    ////////////////////////////////////
    public int addDamegiCharacter( int whr ){
        int ans = lookupWhere( whr );
        if( ans != -1){
            Chctr[ans].addDamegi();
            if( Chctr[ans].damegi() == 0 ){
                Chctr[ans] =new Character( 0,1 );
            }
            return 1;
        }
        return -10;
    }
    //////////////////////////////////////////////
    /* Where定数にwhrと同じ数字を持つ配列の番号を返す */
    /////////////////////////////////////////////
    public int lookupWhere( int whr ){
        int check = -1;
        for(int i=1;i<=Teki;i++)
            check = ( Chctr[i].where() == whr )?i:check;
        return check;
    }
    //////////
    /* 更新 */
    /////////
```

★わっ！おどろいたでしょ！？（滋賀県八日市市・GUNDOLL32歳）……【影：わっ、びっくりした―。おどかさないでよ。つぐ美：えっ？私はなにも？？影：暑さで化粧が流れ落ちてるよ。つぐ美：えっ！やだー、この顔で電車乗ってたのよ。うわーん。】

★Pentium4パソコンで、Windows98の環境を作ろうとしているが、なかなかうまくいかない。Windows2000だとあっさり構築できたのに……。(福岡県北九州市・環31歳) ……**【編：うちの会社では、PCでトラブルがあると、ペーマガ編集部が頼りにされてます。】**

```
    public int update( int key[] ){
        int ans = 0;
        for(int i=1;i < key.length;i++){
            if( key[i] == 1 )ans += addDamegiCharacter( i );
            key[i] = 0;
        }
        for(int i=1;i<=Teki;i++){
            if( Chctr[i].type() != 0 )
                Out = ( Chctr[i].update() )?true:Out;
            else addTeki( i );
        }
        return ans;
    }
    /////////
    /* 描画 */
    /////////
    public void paint(Graphics g){
        for(int i = 1;i < MAX_CHARA;i++)
            Chctr[i].paint( g,Main.Img[ Chctr[i].type() ],Comp );
        if( Out )Chctr[0] =new Character( 4,5 );
        Chctr[0].paint( g,Main.Img[ Chctr[0].type() ],Comp );
    }
    //////////////////////////
    /* GAMEOVERの判定を返す。 */
    //////////////////////////
    public boolean getOut(){
        return Out;
    }
    ///////////////////////
    /* 現在のレベルを返す */
    ///////////////////////
    public int getLevel(){
        return Teki;
    }
}
```

《リスト3》GameCanvas.javaのプログラム・リスト

```
import java.awt.*;
import java.awt.event.*;

/* ゲーム */
public class GameCanvas extends Canvas implements Runnable,MouseListener,KeyListener{
    //////////////////
    /* Final Option */
    //////////////////
    final int GAME_SPEED = 61;          //ゲームスピード
    final int FREESE_SPEED = 2500;      //フリーズスピード
    ////////////
    /* Option */
    ////////////
    boolean LOOP,                       //ソメッド管理用
            ON_PLAY,                    //ゲーム中
            GAMEOVER;                   //ゲームオーバー
    int Speed,                          //ゲームスピード
        Sccore,                         //スコア
        Wid,                            //画面の幅
        Hei;                            //画面の高さ
    int Key[] = {0,0,0,0,0,0,0,0,0,0};  //キー判定
    String Plsstr,                      //スタート前に表示するＳｔｒｉｎｇ
           Tostr;                       //スタート前に表示すｂｒｔｒｉｎｇ
    CharacterManager ChrctrMngr;        //キャラクターマネージャ
    Thread T;                           //スレッド
    ////////
    /* new */
    ////////
    public GameCanvas( int wid,int hei ){
        Speed = GAME_SPEED;
        LOOP = true;
        Wid = wid;
        Hei = hei;
        addMouseListener( this );
        addKeyListener( this );
        newGame();
    }
    ////////////
    /* new Game*/
    ////////////
    public void newGame(){
        Sccore = 0;
        Plsstr ="Please Click";
        Tostr  ="to Start!!";
        ChrctrMngr = new CharacterManager( this );
        ON_PLAY  = false;
        GAMEOVER = false;
    }
    //////////
    /* start */
    //////////
    public void start(){
        T = new Thread(this);
        T.start();
    }
    ////////
    /* run */
    ////////
    public void run(){
         while( LOOP ){
             try{
                 if( GAMEOVER ){
                     removeMouseListener( this );
                     removeKeyListener( this );
                     LOOP = false;
                     Main.Managar = Main.MS_SCCR;
                 }
                 if( ON_PLAY && !GAMEOVER )
                     Sccore += ChrctrMngr.update( Key );
                     Sccore = ( Sccore > 0 )?Sccore:0;
                 if( ChrctrMngr.getOut() ){
                     ON_PLAY = false;
                     GAMEOVER = true;
                     Speed = FREESE_SPEED;
                 }
                 repaint();
                 Thread.sleep( Speed );
             }catch(Exception e){}
         }
    }
    ////////////
    /* destroy */
    ////////////
    public void destroy(){
        LOOP = false;
    }
    /////////////////
    /* To Bufferring */
    /////////////////
    public void update(Graphics g){
         paint( g );
    }
    //////////
    /* paint */
    //////////
    public void paint( Graphics g ){
        Dimension d = getSize();
        Image Bffr = createImage( Wid,Hei );
        Graphics Bffrg = Bffr.getGraphics();
```

```
        drawGraphics( Bffrg );

        g.drawImage( Bffr,0,0,Wid,Hei,this );
    }
    //////////////
    /* View Game */
    //////////////
    public void drawGraphics(Graphics g){
        g.drawImage( Main.Img[0],0,0,this );
        ChrctrMngr.paint( g );
        DrawString( g );
    }
    ////////////////
    /* View Sccore */
    ////////////////
    public void DrawString(Graphics g){
        String scr = "" + Sccore,
               lvl = "Level " + ChrctrMngr.getLevel();
        Font fnt = g.getFont();
        FontMetrics fm  = g.getFontMetrics( fnt );
        int wid = fm.stringWidth( scr );

        g.setColor(Color.white);
        g.drawString(lvl,5,10);
        g.drawString(scr,Wid-5-wid,10);
        wid = fm.stringWidth( Plsstr );
        g.drawString( Plsstr,(Wid-wid)/2,Hei-30 );
        wid = fm.stringWidth( Tostr );
        g.drawString( Tostr,(Wid-wid)/2,Hei-20 );
    }
    ///////////////////
    /* MouseListener  */
    ///////////////////
    public void mouseClicked(MouseEvent me){
        Plsstr = "";
        Tostr = "";
        ON_PLAY = true;
    }
    public void mouseEntered(MouseEvent me){}
    public void mouseExited(MouseEvent me){}
    public void mousePressed(MouseEvent me){}
    public void mouseReleased(MouseEvent me){}
    ////////////////
    /* KeyListener */
    ///////////////
    public void keyReleased(KeyEvent ke){    }
    public void keyPressed(KeyEvent ke){    }
    public void keyTyped(KeyEvent ke){
        char ky = ke.getKeyChar();
        if( ON_PLAY ){
            switch( ky ){
                case '1':Key[1] = 1; break;
                case '2':Key[2] = 1; break;
                case '3':Key[3] = 1; break;
                case '4':Key[4] = 1; break;
                case '6':Key[6] = 1; break;
                case '7':Key[7] = 1; break;
                case '8':Key[8] = 1; break;
                case '9':Key[9] = 1; break;
                default :break;
            }
        }
    }
}
```

《リスト4》Main.javaのプログラム・リスト

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.net.*;

/* メイン */
public class Main extends Applet implements Runnable{
    /////////////////
    /* Final Option */
    /////////////////
    final int BASE_SPEED = 100;               //ベーススピード
    static final int MS_OPEN = 100;           //オープニングメッセージ
    static final int MS_GAME = 101;           //ゲームメッセージ
    static final int MS_SCCR = 102;           //スコアメッセージ
    static final int MS_NEWGAME = 103;        //ニューゲームメッセージ
    //////////////////
    /* Static Option */
    /////////////////
    static OpenCanvas   OpCnvs;               //オープニング
    static GameCanvas   GmCnvs;               //ゲーム
    static SccorePanel  ScPanl;               //スコア
    static Image Img[] = new Image[10];       //表示用のイメージ
    static int Managar;                       //メッセージを受け取る
    ///////////
    /* Option */
    ///////////
    boolean LOOP;                             //ソメッド管理用
    int Speed;                                //ゲームスピード
    Thread T;                                 //スレッド
    /////////
    /* init */
    /////////
    public void init(){
        LOOP = true;
        Speed = BASE_SPEED;
        String ImageName[] = { "Image/map.gif",
                               "Image/roketto.gif",
                               "Image/nuu011.gif",
                               "Image/nuu021.gif",
                               "Image/baku.gif",
                               "Image/nuu001.gif",
                               "Image/nuu002.gif",
                               "Image/nuu003.gif"
                             };
        LoadImage( this.getCodeBase(),ImageName,Img );
        NewGame();
        start();
    }
    public void NewGame(){
        Dimension d = getSize();
        Managar = MS_OPEN;
        OpCnvs = new OpenCanvas( d.width,d.height );
        GmCnvs = new GameCanvas( d.width,d.height );
        ScPanl = new SccorePanel( d.width,d.height );
        OpCnvs.setBounds( 0,0,d.width,d.height );
        GmCnvs.setBounds( 0,0,d.width,d.height );
        ScPanl.setBounds( 0,0,d.width,d.height );
        setLayout(null);
    }
    //////////
    /* start */
    //////////
    public void start(){
        T = new Thread( this );
        T.start();
    }
    ////////
    /* run */
    ////////
    public void run(){
        while( LOOP ){
            try{
```

★さすがベーマガ！いろいろなモノのスリム化にならって、本誌もスリムになるとは（笑）！！（宮崎市・810 24歳）……【影：時代を先取りするベーマガとしては、世の流れにそって、本誌もスケルトンにしようかと思ってます。みなさんのご意見を。編：ブー。】

★今年の中日はダメですね，編さん。後半に期待しましょう。（愛知県名古屋市・ドラゴンズファン2号20歳）……【影：編くんがドラゴンズファンだって知ってるとは。編：いよいよペナントレースも後半戦，まあ，来年につながる戦いをしてもらえればいいや。】

```
                Serect();
                Thread.sleep( Speed );
            }catch( Exception e ){}
        }
    }
    ////////////
    /* destroy */
    ////////////
    public void destroy(){
        LOOP = false;
    }
    //////////
    /* paint */
    //////////
    public void paint(){
    }
    ///////////
    /* Serect */
    ///////////
    public void Serect(){
        if( Managar != 0 ){
            switch( Managar ){
                case MS_OPEN :
                    remove( GmCnvs );
                    remove( ScPanl );
                    add( OpCnvs );
                    OpCnvs.start();
                    Managar = 0;
                    break;
                case MS_GAME :
                    remove( OpCnvs );
                    remove( ScPanl );
                    add( GmCnvs );
                    GmCnvs.start();
                    Managar = 0;
                    break;
                case MS_SCCR :
                    remove( OpCnvs );
                    remove( GmCnvs );
                    ScPanl.setScorre( GmCnvs.Sccore );
                    add( ScPanl );
                    Managar = 0;
                    break;
                case MS_NEWGAME :
                    remove( OpCnvs );
                    remove( GmCnvs );
                    remove( ScPanl );
                    NewGame();
                    break;
                default :
                    break;
            }
        }
    }
    ///////////////
    /* Load Image */
    ///////////////
    public int LoadImage( URL url,String imgstr[],Image img[] ) {
        int i,size = img.length;
        MediaTracker tracker = new MediaTracker( this );

        for( i = 0;i < imgstr.length;i++ ){
            if( i < size ){
                img[i] = this.getImage( url,imgstr[i] );
                tracker.addImage( img[i],i );
            }
        }
        while( ! tracker.checkAll()) {
            try { tracker.waitForAll(); }
            catch(InterruptedException e) { return -1; }
            }
        return i;
    }
}
```

《リスト5》OpenCanvasのプログラム・リスト

```
import java.awt.*;
import java.awt.event.*;

/* オープニング */
public class OpenCanvas extends Canvas implements Runnable,MouseListener{
    //////////////////
    /* Final Option */
    //////////////////
    final int OPEN_SPEED = 401;    //オープニングスピード
    final int FREESE_SPEED = 2500; //フリーズスピード
    ////////////
    /* Option */
    ////////////
    boolean LOOP;                  //ソメッド管理用
    int Speed,                     //ゲームスピード
        OpnImg,                    //表示イメージの番号
        Wid,                       //画面の幅
        Hei;                       //画面の高さ
    Font Fn[]     = new Font[3];   //フォント設定
    String Str[] = new String[3];  //文字設定
    Thread T;                      //スレッド
    ////////
    /* new */
    ////////
    public OpenCanvas( int wid,int hei ){
        Speed = OPEN_SPEED;
        LOOP = true;
        OpnImg = 5;
        addMouseListener( this );
        Wid = wid;
        Hei = hei;
        Fn[0]  = new Font( "Courier",Font.BOLD,18 );
        Fn[1]  = new Font( "Courier",Font.ITALIC,11 );
        Fn[2]  = new Font( "Courier",Font.BOLD,11 );
        Str[0] = "ヌー星人";
        Str[1] = "～ロケット救出編～";
        Str[2] = "click  start";
    }
    //////////
    /* start */
    //////////
    public void start(){
        T = new Thread(this);
        T.start();
    }
    ////////
    /* run */
    ////////
    public void run(){
        while( LOOP ){
            try{
                if( OpnImg == 7 ){
                    destroy();
                    removeMouseListener( this );
                    Main.Managar = Main.MS_GAME;
                }else
                    OpnImg = ( OpnImg == 5 )?6:5;
                if( Speed == FREESE_SPEED )
                    OpnImg = 7;
                repaint();
                Thread.sleep( Speed );
            }catch(Exception e){}
```

```
        }
    }
    ////////////
    /* destroy */
    ////////////
    public void destroy(){
        LOOP = false;
    }
    //////////////////
    /* To Bufferring */
    //////////////////
    public void update(Graphics g){
         paint( g );
    }
    //////////
    /* paint */
    //////////
    public void paint( Graphics g ){
        Image Bffr = createImage( Wid,Hei );
        Graphics Bffrg = Bffr.getGraphics();

        drawGraphics( Bffrg );

        g.drawImage( Bffr,0,0,Wid,Hei,this );
    }
    /////////////////
    /* drawGraphics */
    /* 描画用       */
    /////////////////
    public void drawGraphics(Graphics g){
        g.setColor( Color.orange );
            g.fillRect( 0,0,Wid,Hei );
        g.setColor( Color.black );
            My_drawString( g,Fn[0],Str[0],13,Hei / 5);
            My_drawString( g,Fn[1],Str[1],0,Hei / 5 + 10);
        g.drawImage( Main.Img[OpnImg],16,Hei / 5 * 2,this );
        g.setColor( Color.red );
            My_drawString( g,Fn[2],Str[2],10,Hei );
    }
    ////////////////////////////////
    /* Font設定してdrawStringを実行*/
    ////////////////////////////////
    public void My_drawString( Graphics g,Font fn,String st,int x,int y ){
        g.setFont( fn );
        g.drawString( st,x,y );
    }
    ///////////////////
    /* MouseListener  */
    ///////////////////
    public void mouseClicked(MouseEvent me){
        Speed = ( Speed == OPEN_SPEED )?FREESE_SPEED:Speed;
    }
    public void mouseEntered(MouseEvent me){}
    public void mouseExited(MouseEvent me){}
    public void mousePressed(MouseEvent me){}
    public void mouseReleased(MouseEvent me){}
}
```

《リスト6》SccorePanel.javaのプログラム・リスト

```
import java.awt.*;
import java.awt.event.*;

/* スコア結果 */
public class SccorePanel extends Panel implements ActionListener{
    //////////////////
    /* Final Opetion */
    //////////////////
    final int FONTSIZE = 12;    //フォントサイズ
    final int SPACE    = 5;     //スペースサイズ
    final int BUTTONSIZE = 40;  //ボタンサイズ
    final int CONSIZE = 20;     //ボタンサイズ
    /////////////
    /* Opetion */
    /////////////
    Label  Lbl_1,Lbl_2,Lbl_3;   //ラベル
    Button Btn_OK;              //ボタン
    int Scr = 0;                //表示用のスコア
    ////////
    /* new */
    ////////
    public SccorePanel( int wid,int hei ){
        Font fn  = new Font( "Courier",Font.BOLD,FONTSIZE );
        setBackground( Color.orange );

        int y = SPACE;
        Lbl_1 = new Label( "",Label.CENTER );
        Lbl_1.setFont( fn );
        Lbl_1.setForeground( Color.red );
        Lbl_1.setBounds( 0,y,wid,FONTSIZE );

        y += FONTSIZE;
        Lbl_2 = new Label( "高得点目指して",Label.CENTER );
        Lbl_2.setFont( fn );
        Lbl_2.setBounds( 0,y,wid,FONTSIZE );

        y += FONTSIZE;
        Lbl_3 = new Label( "がんばばろう",Label.CENTER );
        Lbl_3.setFont( fn );
        Lbl_3.setBounds( 0,y,wid,FONTSIZE );

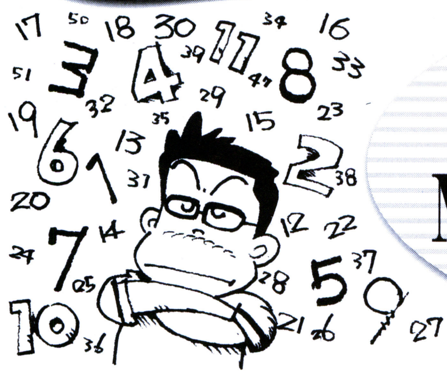
        y += CONSIZE + SPACE;
        Btn_OK = new Button( "OK" );
        Btn_OK.setBounds( (wid - BUTTONSIZE)/2,y,BUTTONSIZE,CONSIZE );
    }
    //////////
    /* paint */
    //////////
    public void paint( Graphics g ){
        add( Lbl_1 );
        add( Lbl_2 );
        add( Lbl_3 );
        add( Btn_OK );
        Btn_OK.addActionListener( this );
    }
    public void setScorre( int scr ){
        Lbl_1.setText( scr + "点です" );
        Scr = scr;
    }
    //////////////////
    /* ActionListener */
    //////////////////
    public void actionPerformed( ActionEvent ae ){
        String str = ae.getActionCommand();
        if( str == "OK" ){
            Btn_OK.removeActionListener( this );
            Main.Managar = Main.MS_NEWGAME;
        }
    }
}
```

《リスト7》HTMLファイルのリスト例

```
<applet code = "Main.class" width = 100 height = 100>
</applet>
```

★ゴールデンウィークは宿題が多すぎた。きっと夏休みはもっと宿題が多いんだろうなあ。英語のレポートが一番いやだー。（神奈川県川崎市・one stop19歳）……【つく美：大学生ですか。勉強と遊びと趣味をバランスよくすることが大事よ。がんばってね。】

Windows(QBasic)用



COMを交えた4人対戦も可能

移植版

Mouse対応版SHORTEN（ショウテン）

tayosan(17歳)

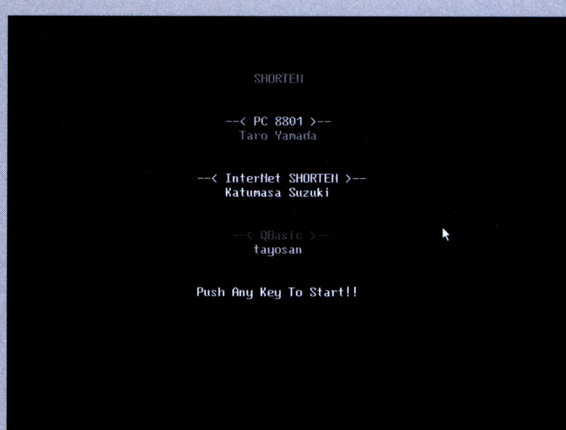
はじめに

本作品はベーマガ掲載の『SHORTEN』(山田太郎さん作　91年2月号掲載)，移植した作品『INTERNET SHORTEN』(鈴木克昌さん作2000年10月号掲載)を，僕が適当にN88-BASICに移植して，それをもとにQBasicに移したものです。

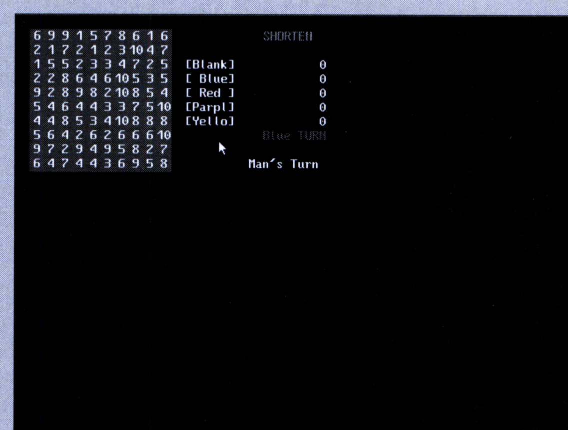
また，これはベーマガ2000年4月号掲載のQBasic用『マウス操作関数群(竹田良之さん作)』を使用しております。

ルール

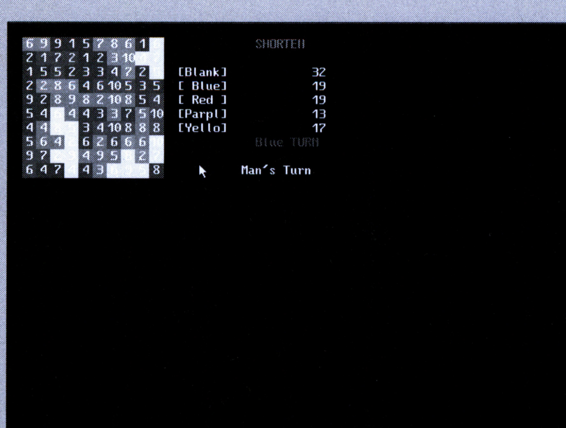
10×10でならべられたカードの一つを交代で選択し，隣接する（上下左右）カードの値より選択したカードの値のほう



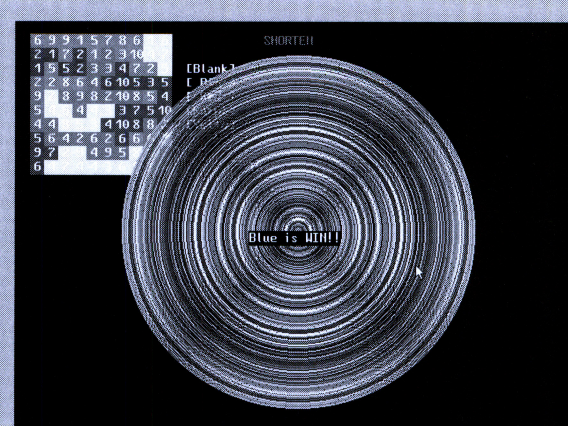
《写真１》Qbasicへの移植された人気投稿作品



《写真２》マウスへの対応がウレシイ機能



《写真３》COMを交えた対戦はかなり白熱



《写真４》ルールも簡単なので手軽に遊べる

★メインPC修理中のなか，サブマシン1も故障した。サブマシン2は遅いし動作不安定で最悪だ！（北海道空知郡・WILDWOLF19歳）……【影：壊れるときは，連鎖的に，なにもかもダメってこと。あー！私のPCも故障した！サブマシンっと。編：それ，私のですよ。】

が大きければ自分のポイントになります。未選択のカードがなくなった時点でポイントの高いプレイヤーが勝ちとなります(全文引用)。

ゲームの開始

プログラムを実行して何かキーを押すと対戦相手を決めます。キーボード操作専用にすると，1=人間，2=CPUで，マウス専用にするとLeft=人間，Right=CPUでBlueが先手になります。言い忘れていましたが，ゲームがDRAWのときは後手の勝ちになります。

改造

このプログラムでは，盤の最大横幅(BanMaxX)，盤の最大縦幅(banmaxy)，カードの最大値(nummax)，操作モード(MsOpt，0=キーボード,1=マウス使用)の値を変えることで色々な設定で遊べます。

省略したい人

長〜いと思う人はどちらか一つに絞って入力するとよいでしょう。できる限り短くするならマウスドライバを切り離すとよいでしょう。MsOpt=0としてMsg以外のサブやファンクションを切る。また，IF文でMsOptを比較しているところも省略できますが，そこは本人の責任において変更してください。

感想

このゲームはCPUの強さを求めたつもりなんだけど「俺より弱い」。初心者の俺に負けるようではねぇ…。まぁ，ないよりはましかな？マウス対応は難しいね!!

マイHPには未対応版も置こうと思っています。感想がある人は，tayosan@tokutoku.zzn.comまで送ってね!!

CHECKER FLAG

編：基本的なゲームシステムの変更などはありませんが，マウス対応やCOMを交えた4人対戦など,遊びかたの面での機能が拡張されている作品です。

つぐ美：ルールが単純でありながら，対戦における駆け引きが非常に熱いゲームです。COMとの対戦もいいけど友達同士で遊ぶよりいっそう面白いんじゃないかしら？

Dr.D：リスト面では，以前紹介した『マウス操作関数群』をしっかり利用しているのが目に付くのぉ。自分が有益だと思ったライブラリ投稿などは，どんどん使っていくほうが自分の作品も進化していくし，そのライブラリを作成した原作者もウレシイじゃろう。

『Mouse対応版SHORTEN』のプログラム・リスト

```
DECLARE SUB SetMsPos (XPos AS INTEGER, YPos AS INTEGER)
'-------------------------------<< Ishokuban [ SHORTEN ] (tayosan) >>--------
DECLARE SUB Msg (st AS STRING, y!, C AS INTEGER)
DECLARE FUNCTION SYSINT! (NUM%, AX%, BX%, CX%, DX%, SI%, DI%, ES%)
DECLARE FUNCTION Msinit! (BTN AS INTEGER)
DECLARE FUNCTION GetMsSts! (BTN AS INTEGER, XPos AS INTEGER, YPos AS INTEGER)
DECLARE SUB MsOn ()
DECLARE SUB MsOff ()

SCREEN 12: CLS : RANDOMIZE TIMER: MsOpt = 1: PRINT "Please Wait a moment"
DIM BanMaxX, BanMaxY, NumMax, player, Nin, w, w1, w2 AS INTEGER, In$
FOR i = 0 TO 2: t$ = TIME$: WHILE t$ = TIME$: w = w + SGN(i) * .01: WEND: NEXT
WHILE Nin = 0: CLS : INPUT "Please Input Player(2-4)"; Nin
IF Nin < 2 OR Nin > 4 THEN BEEP: Nin = 0
WEND: CLS : BanMaxX = 10: BanMaxY = 10: NumMax = 10: Cm = BanMaxX * BanMaxY
DIM StCh%(320, NumMax), CX(Cm), CY(Cm), CN(Cm), Gtc(Nin), Pl(Nin) AS INTEGER
DIM BanX, BanY AS INTEGER
DIM Ts(7) AS STRING, Ty(7), Tc(7) AS INTEGER
DIM BanNum(BanMaxX, BanMaxY) AS INTEGER
DIM BanPly(BanMaxX, BanMaxY) AS INTEGER
FOR i = 0 TO 7: READ Ts(i), Ty(i), Tc(i): NEXT
FOR i = 1 TO NumMax: PRINT i: GET (8, 0)-(24, 16), StCh%(0, i): CLS : NEXT
'----------------------------------------------------------------------------
DATA SHORTEN,5,12,--< PC 8801 >--,8,11,Taro Yamada,9,13
DATA --< InterNet SHORTEN >--,12,14,Katumasa Suzuki,13,15
DATA --< QBasic >--,16,9,tayosan,17,10,Push Any Key To Start!!,20,15
'----------------------------------------------------------------------------
Title: PLAY "l64t255cc+dd+eff+gg+aa+b": IF MsOpt = 1 THEN GOSUB MsMove
CLS : FOR i = 0 TO 7: Msg Ts(i), Ty(i), Tc(i): NEXT
IF MsOpt = 1 THEN A% = Msinit(B%): CALL MsOn: A% = GetMsSts(B%, C%, D%)
GOSUB KW: GOTO ft
'----------------------------------------------------------------------------
KW: In$ = "": WHILE In$ = "": In$ = INKEY$
IF MsOpt = 1 THEN WHILE B% <> 0: A% = GetMsSts(B%, C%, D%): WEND
IF In$ = CHR$(27) THEN
  IF MsOpt = 1 THEN CALL MsOff
  END
END IF
IF MsOpt = 1 THEN A% = GetMsSts(B%, C%, D%): IF B% <> 0 THEN In$ = "."
WEND: RETURN
'----------------------------------------------------------------------------
ft: IF MsOpt = 1 THEN GOSUB MsMove
CLS : Msg "Player Set", 3, 12: FOR i = 0 TO Nin - 1
S$ = MID$("BlueRed ParpYell", i * 4 + 1, 4)
  IF MsOpt = 0 THEN
    A% = 0: WHILE A% = 0
```

★ロボ道場かどこかで，テスターの基礎的な使いかたを教えていただけませんか？普段は電池の残量を計るくらいしか使ってないんで。保存版のカードか何かにしてもらえるとうれしいんですが。(奈良市・池田　博42歳）……【編：電子工作的な記事も要望多いです。】

```
      Msg S$ + " (Player=1,Commputer=2)>", 5 + i, 15
      INPUT P1(i): IF P1(i) < 1 OR P1(i) > 2 THEN BEEP ELSE A% = 1
    WEND
  ELSE
    Msg S$ + " (Player=Left,Commputer=Right)", 5 + i, 15
    F1 = 0: WHILE F1 = 0: GOSUB KW: IF B% > 0 AND B% < 3 THEN F1 = 1
    WEND: P1(i) = B%
  END IF
NEXT: player = 0: CLS
'--------------------------------------------------------------------------
setdat: FOR i = 1 TO BanMaxX: FOR j = 1 TO BanMaxY
  BanNum(j, i) = INT(RND * NumMax) + 1: BanPly(i, j) = 0
NEXT j, i
'--------------------------------------------------------------------------
main: GOSUB display
Msg MID$("BlueRed ParpYell", player * 4 + 1, 4) + " TURN", INT((BanMaxY) / 2) + 4, VAL(MID$
("09121314", player * 2 + 1, 2))
SELECT CASE P1(player)
  CASE 1
    PLAY "l32t120fefe"
    Msg "Man's Turn ", INT((BanMaxY) / 2) + 6, 15
    Mx = INT(BanMaxX / 2 + .5): My = INT(BanMaxY / 2 + .5)
    F1 = 0
    IF MsOpt = 1 THEN
      WHILE F1 = 0
        B% = 0
        WHILE B% = 0
          A% = GetMsSts(B%, C%, D%)
        WEND
        Mx = INT(C% / 16): My = INT(D% / 16)
        IF Mx >= 1 AND Mx <= BanMaxX AND My >= 1 AND My <= BanMaxY THEN
          IF BanPly(Mx, My) = 0 THEN F1 = 1
        END IF
      WEND
    ELSE
      WHILE F1 = 0
        In$ = "": WHILE In$ = "": In$ = INKEY$
          FOR i = 0 TO 2
            LINE (Mx * 16 + 1, My * 16 + 1)-STEP(14, 14), 11, B
          NEXT
          LINE (Mx*16+1,My*16+1)-STEP(14,14),VAL(MID$("0809121314",BanPly(Mx, My)*2+1,2)),B
        WEND
        Mx=Mx+(In$="4")-(In$="6"): IF Mx<1 THEN Mx=1 ELSE IF Mx>BanMaxX THEN Mx=BanMaxX
        My=My+(In$="8")-(In$="2"): IF My<1 THEN My=1 ELSE IF My>BanMaxY THEN My=BanMaxY
        IF (In$ = " " OR In$ = CHR$(13)) AND BanPly(Mx, My) = 0 THEN F1 = 1
      WEND
    END IF
  CASE 2
    PLAY "t120l32cde"
    Msg "Thinking...", INT((BanMaxY) / 2) + 6, 14
    ck = 0: FOR i = 1 TO BanMaxX: FOR j = 1 TO BanMaxY: cp = 0
    IF BanPly(i, j) = 0 THEN
      CX(ck) = i: CY(ck) = j: CN(ck) = BanNum(i, j)
      IF i - 1 >= 1 THEN
        IF BanNum(i - 1, j) < CN(ck) THEN cp = cp + 1
        IF BanNum(i - 1, j) >= CN(ck) - 3 THEN cp = cp + 1
      END IF
      IF i + 1 <= BanMaxX THEN
        IF BanNum(i + 1, j) < CN(ck) THEN cp = cp + 1
        IF BanNum(i + 1, j) >= CN(ck) - 3 THEN cp = cp + 1
      END IF
      IF j - 1 >= 1 THEN
        IF BanNum(i, j - 1) < CN(ck) THEN cp = cp + 1
        IF BanNum(i, j - 1) >= CN(ck) - 3 THEN cp = cp + 1
      END IF
      IF j + 1 <= BanMaxY THEN
        IF BanNum(i, j - 1) < CN(ck) THEN cp = cp + 1
        IF BanNum(i, j - 1) >= CN(ck) - 3 THEN cp = cp + 1
      END IF
      CN(ck) = CN(ck) + cp
      ck = ck + 1
    END IF
    NEXT j, i
    FOR i = NumMax + 8 TO 1 STEP -1: FOR j = 0 TO ck - 1
      IF CN(j) = i THEN Mx = CX(j): My = CY(j): GOTO sec
    NEXT j, i
sec: GOSUB display: GOSUB wt: END SELECT
nb = BanNum(Mx, My): BanPly(Mx, My) = player + 1
IF Mx - 1 >= 1 THEN IF BanNum(Mx - 1, My) < nb THEN BanPly(Mx - 1, My) = player + 1
IF Mx + 1 <= BanMaxX THEN IF BanNum(Mx + 1, My) < nb THEN BanPly(Mx + 1, My) = player + 1
IF My - 1 >= 1 THEN IF BanNum(Mx, My - 1) < nb THEN BanPly(Mx, My - 1) = player + 1
```


★僕の通っている本屋にはベーマガが1冊しかありません。つまり僕のためだけです。これじゃベーマガ読者が増えません。どうすりゃいいんだ！（兵庫県神戸市・JANDY18歳）……【影：ベーマガ読者なら，その本屋に直接お願いしなきゃ。お願いしてください。】

```
IF My + 1 <= BanMaxY THEN IF BanNum(Mx, My + 1) < nb THEN BanPly(Mx, My + 1) = player + 1
FOR i = 0 TO Nin: Gtc(i) = 0: NEXT
FOR i = 1 TO BanMaxX: FOR j = 1 TO BanMaxY: nb=BanPly(i,j): Gtc(nb)=Gtc(nb)+1: NEXT j, i
IF Gtc(0) = 0 THEN GOTO ed
player = player + 1: IF Nin = player THEN player = 0
GOTO main
'---------------------------------------------------------------------------
display: IF MsOpt = 1 THEN GOSUB MsMove
Msg "SHORTEN", 2, 12: FOR i = 1 TO BanMaxX: FOR j = 1 TO BanMaxY
  LINE (i * 16, j * 16)-STEP(16, 16), VAL(MID$("0809121314", BanPly(i, j) * 2 + 1, 2)), BF
  PUT (i * 16 - (BanNum(i, j) < 10) * 4, j * 16), StCh%(0, BanNum(i, j)), OR
NEXT j, i: FOR i = 0 TO Nin: LOCATE INT(BanMaxY / 2 - Nin / 2 + 1) + i, BanMaxX * 2 + 5
PRINT "["; MID$("Blank Blue Red ParplYello", i * 5 + 1, 5); "]", Gtc(i): NEXT: RETURN
'---------------------------------------------------------------------------
ed: GOSUB display: FOR i = 0 TO 200: CIRCLE (320, 240), i, RND * 16: NEXT: win = 0
j = 0: FOR i = 1 TO Nin: IF Gtc(i) >= j THEN j = Gtc(i): win = i - 1
NEXT: Msg MID$("BlueRed ParpYell", win*4+1, 4) + " is WIN!!", 16, 15: PLAY "t120l4gfedc"
GOSUB display: GOSUB KW: GOTO Title
'---------------------------------------------------------------------------
wt: FOR w1 = 0 TO w: FOR w2 = 0 TO w / 5: NEXT w2, w1: RETURN
MsMove: SetMsPos 0, 0: RETURN

FUNCTION GetMsSts (BTN AS INTEGER, XPos AS INTEGER, YPos AS INTEGER)
    F1% = SYSINT(&H33, 3, BTN, XPos, YPos, 0, 0, 0)
    GetMsSts = F1%
END FUNCTION

SUB Msg (st AS STRING, y, C AS INTEGER)
  LOCATE y, 40 - LEN(st) / 2: COLOR C: PRINT st; : COLOR 15
END SUB

FUNCTION Msinit (BTN AS INTEGER)
    AX% = 0
    F1% = SYSINT(&H33, AX%, BTN, 0, 0, 0, 0, 0)
    Msinit = AX%
END FUNCTION

SUB MsMove
    SetMsPos 0, 0
END SUB

SUB MsOff
    F1% = SYSINT(&H33, 2, 0, 0, 0, 0, 0, 0)
END SUB

SUB MsOn
    F1% = SYSINT(&H33, 1, 0, 0, 0, 0, 0, 0)
END SUB

SUB SetMsPos (XPos AS INTEGER, YPos AS INTEGER)
    F1% = SYSINT(&H33, 4, 0, XPos, YPos, 0, 0, 0)
END SUB

FUNCTION SYSINT (NUM%, AX%, BX%, CX%, DX%, SI%, DI%, ES%)
    DIM INTER%(36)
    INTER%(0) = &H68B: INTER%(1) = VARPTR(AX%)
    INTER%(2) = &H1E8B: INTER%(3) = VARPTR(BX%)
    INTER%(4) = &HE8B: INTER%(5) = VARPTR(CX%)
    INTER%(6) = &H168B: INTER%(7) = VARPTR(DX%)
    INTER%(8) = &H5756
    INTER%(9) = &H368B: INTER%(10) = VARPTR(SI%)
    INTER%(11) = &H3E8B: INTER%(12) = VARPTR(DI%)
    INTER%(13) = &H9006
    INTER%(14) = &H68E: INTER%(15) = VARPTR(ES%)
    INTER%(16) = ((NUM% * 2 ^ 8) AND &HFF00) OR &HCD
    INTER%(17) = &H268C: INTER%(18) = VARPTR(FLAG%)
    INTER%(19) = &H68C: INTER%(20) = VARPTR(ES%)
    INTER%(21) = &H9007
    INTER%(22) = &H3E89: INTER%(23) = VARPTR(DI%)
    INTER%(24) = &H3689: INTER%(25) = VARPTR(SI%)
    INTER%(26) = &H5E5F
    INTER%(27) = &H1689: INTER%(28) = VARPTR(DX%)
    INTER%(29) = &HE89: INTER%(30) = VARPTR(CX%)
    INTER%(31) = &H1E89: INTER%(32) = VARPTR(BX%)
    INTER%(33) = &H689: INTER%(34) = VARPTR(AX%)
    INTER%(35) = &HCB
    DEF SEG = VARSEG(INTER%(0))
    CALL ABSOLUTE(VARPTR(INTER%(0)))
    ERASE INTER%
    SYSINT = FLAG%
END FUNCTION
```


★ワンダーウィッチでプログラミング！心に決めて「ワンダーウィンチ」を買ったが，いまだに封印を解いていない。（福島市・ウィッチーさん19歳）……【影：よっぽど惚れ込んでるんだね。汚したくないわけだ。ワンダーウィッチは使ってもらいたいだろうに。】

Windows(QBasic)用

紙飛行機よ永遠なれ，長く飛ばし続けよう！



Paper Plane Forever
（ペーパー プレイン フォーエバー）

山屋　健（21歳）

内　容

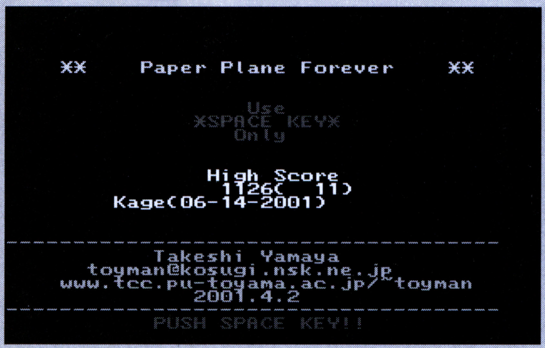
紙飛行機を飛ばしつづけましょう。右端へ行くと100点追加され，左に戻ります。また，ブロックに当たるか，墜落するとゲーム・オーバーです。ちなみに左端から右端には行けません。

《第１表》変数表

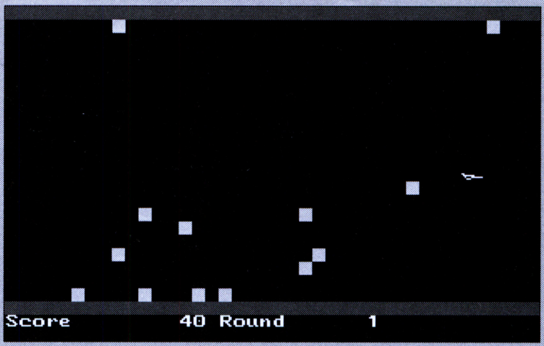
〈飛行機関係〉	
px,py	飛行機の座標
pre.px,pre.py	前回の飛行機の座標
angle	飛行機の角度。64で360度
backangle	飛行機の尾翼の角度
v	飛行機の速さ
〈ブロック関係〉	
bx(32),by(32)	ブロックの座標です
bt(32)	ブロックが落ちるまでの残り時間
bnum	現在移動中のブロックの個数
bmap(41,25)	ブロックがあるときは0以外が入る。判定用
〈ゲーム全体〉	
score	得点
highscore	最高得点
time	経過時間（サイクル数）
round	周回数.右から左へ行った回数
gravity	重力
gamewait	ゲームの速さ

使いかた・遊びかた

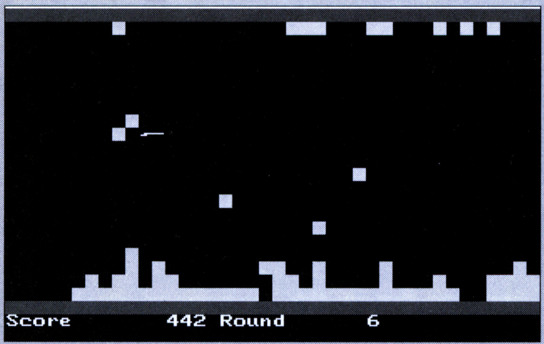
まず，データファイルを作ってください。最初の行のコメント（'）を外して実行するとできます。終わったら，また' をつけてください。スペース・キーでスタートです。



《写真１》スペース・キーだけで遊べるゲーム



《写真２》紙飛行機の飛行体制を整えよう



《写真３》上から降ってくるブロックがくせ者だ



《写真４》これが影さんのハイスコア。まずはこの記録に挑戦だ

★はじめてハガキ出します。編集長は，本当にあんな顔してるんですか？ホームページに顔描いてありました。（新潟県新発田市・なしんぼ20歳）……**【編集長：あんな顔って，どういうことだ？！影：久々に登場したんですから，怒らない，怒らない。】**

《第2表》ラベル名表

initial	初期化しています。スクリーンの設定， 配列の定義，効果音の読み込みをしています
title	タイトル画面です。見たままです
start	変数を初期化しています。 そのあとにカウントダウンの表示もしています
gamemai	メインループです。飛行機のサブルーチンと，ブロックのサブルーチンを呼び，また，終条件の判定，得点の判定をしています
gameover	ゲームオーバーの画面です。 ここでハイスコアの更新も行います
plane	スペース・キーの状態を読み込み，機体の計算をします。 結構怪しい式ですが，それなりに考察の結果です
plane.draw	現在の機体の変数から，ラインを使って機体を描きます。 上半分は，黒で消しています
block.init	ブロックの初期化をします
block.proc	ブロックの生成と，移動を行います。 ブロックは，8ドット単位で移動しています
bock.generate	ブロックの生成です。ランダムを使っています
block.display	ブロックを表示します
block.del	ブロックを一掃します
goal	右端へついたときに呼ばれます。 変数を変更して，ゲームに変化を持たせます。 ここが，長く遊んでもらうための工夫のしどころです
keywait	スペース・キーを押されるのを待ちます
waiter	VSYNCをまって，タイミングを取ります
waiter2	適当な間をとるときにしようしています

操作方法

いつでもスペース・キーのみ使います。以下に細かなテクニックを書きますが，プレイしたほうが速いです。

・速度が重要です。そのときの速度に関して，機体の動きが変わります。

・細かく連打すると加速します。バランスを取りながら，進みましょう。

・押している間，機首が上に上がります。速度が大きければ宙返りをします。これはブロックを避けるときに多用します。

プログラムの仕組み

大きくわけて，紙飛行機の動きを作る部分，紙飛行機を描く部分，ブロックを生成し動かす部分，それらを呼び出すメインループで成り立っています。

ラベル名表で，その部分の説明をしています。

プログラムの入力方法と注意点

変数名を間違えると，エラーも出ないでおかしくなるので注意しましょう。変な動きがするときは，変数を見なおします。飛行機の動きの計算は1か所に固めてあるので，そこを重点的にやります。

改造のアドバイス

backangleをキー入力によって変更していますが，これを変更すると動きが変わって面白いです。また，gravityを変えても動きが大きく変わります。

★ 協力

・富山県立大学テクニカルコンピュータクラブのみなさん

CHECKER FLAG

つぐ美：スペース・キーしか使わないので簡単に遊べるわね。それでいながら紙飛行機の動きが独特なので，すぐに飽きるということはないですね。

Dr.D：その紙飛行機の動きは三角関数などを用いているので，単調な動きにならず，つぐ美クンの言うような独特の動きを実現しておる。LINE命令による紙飛行機の描画も座標に合わせた形式にしてあるため，見た目は簡素ながらもいろいろな体制で描かれているのが目に付くはずじゃ。あとは，単純な形ながらもファイル処理を使ってハイスコア保存をしておるのはQBasic投稿では珍しいのではないかな？専門的な部分では，INPを利用してVSYNC(垂直同期)を待つルーチンを作成しているのは，こだわっている部分だと感じたゾ。

『Paper Plane Forever』のプログラム・リスト

```
'
'                    Paper Plane
'
'                    2001-4 Takeshi Yamaya(toyman@kosugi.nsk.ne.jp)
'

'GOTO makedatafile

initial: '---------------------------------------------
  SCREEN 7: RANDOMIZE INT(TIMER)
  CLS : PLAY "mb t180"
  'constance set
  pi! = 3.141592 / 32: gamewait = 5: gravity = -.8
  OPEN "pplane.dat" FOR INPUT AS #1
  INPUT #1, a$: highscore = VAL(a$):
  INPUT #1, a$: highround = VAL(a$):
  INPUT #1, name$
  CLOSE
  DIM bmap(41, 25) AS INTEGER
  DIM bx(32) AS INTEGER, by(32)  AS INTEGER, bt(32) AS INTEGER
  DIM pre.bx(32) AS INTEGER,.pre.by(32) AS INTEGER
  DIM se$(10): RESTORE

  FOR i = 0 TO 7: READ se$(i): NEXT
  DATA "mf o2 l16cgeg>cgcg>egcg>egce"
  DATA "mfo3l16c>cp5"
  DATA "mb o1l32c"
  DATA "mbo3c32"
  DATA "mbo4g32"
  DATA "mf o4l8cgeb>p8cgc"
  DATA "mb l32o1cgdc>>cc<<<ggg>>>cc<<<gcgg"
  DATA "mb o4l16gcg"
 title: '--------------------------------------------
  CLS
  LOCATE 5, 1
  COLOR 10

  PRINT : PRINT :
  PRINT "    **    Paper Plane Forever    **  "
  PRINT : PRINT :
  COLOR 5:
  PRINT "                 Use                "
  PRINT "             *SPACE KEY*            "
  PRINT "                Only                "
  PRINT : PRINT :
  COLOR 14
  PRINT "              High Score "
  PRINT USING "       #############(####)"; highscore; highround
  PRINT "          "; name$
  COLOR 7
  PRINT : PRINT :
  PRINT "------------------------------------"
  PRINT "             Takeshi Yamaya         "
  PRINT "        toyman@kosugi.nsk.ne.jp     "
  PRINT "     www.tcc.pu-toyama.ac.jp/ toyman "
  PRINT "               2001.4.2             "
  PRINT "------------------------------------"
  PLAY se$(7)
  GOSUB keywait

 start: '---------------------------------------------
  px! = 150: py! = 50: vx! = 0: vy! = 0
  pre.px = px: pre.x1 = x1: pre.x2 = x2
```

★ところで，つぐ美ちゃん以外の人の年齢は？（千葉県佐倉市・**野上健太16歳**）……【**Dr.D**：ワシはある時期から，年を数えるのやめたわい。**影**：というより，数えられなくなったんですよね。**Dr.D**：くー！じゃがプログラムでは若いもんにはまだまだ負けんぞ。】

```
pre.py = py: pre.yl = yl: pre.y2 = y2
w = 0: v = 7: angle = 0: backangle = 2
flyflag = 0: score = 0: round = 0: time = 0
gamewait = 5: gravity = -.8
'make screen
GOSUB block.init

GOSUB block.display
GOSUB plane.draw

'----Are you Ready?
LOCATE 5, 13: PRINT "Are You Ready?"
GOSUB keywait
LOCATE 5, 13: PRINT "                    "

'----count down
FOR j = 3 TO 0 STEP -1
  IF j = 0 THEN PLAY se$(4) ELSE PLAY se$(3)
  FOR i = 10 TO 0 STEP -1
    LOCATE 5, 20: PRINT USING "#"; j
    GOSUB waiter
  NEXT
NEXT
LOCATE 5, 20: PRINT " "


gamemain: '---------------------------------------------
  time = time + 1

  GOSUB plane

  'end check
  IF py > 180 OR py < 8 THEN GOTO gameover
  tmpx = INT(px / 8): tmpy = INT(py / 8)
  IF bmap(tmpx, tmpy) <> 0 THEN GOTO gameover

  'add score
  IF bmap(tmpx - 1, tmpy) <> 0 THEN score = score + 7: PLAY se$(4)
  IF bmap(tmpx, tmpy - 1) <> 0 THEN score = score + 8: PLAY se$(4)
  IF bmap(tmpx + 1, tmpy) <> 0 THEN score = score + 9: PLAY se$(4)
  IF bmap(tmpx, tmpy + 1) <> 0 THEN score = score + 6: PLAY se$(4)


  GOSUB block.proc


 '----Draw
  GOSUB plane.draw
  LOCATE 24, 1: PRINT USING "Score ######### Round ######"; score; round;

  GOSUB waiter
  GOTO gamemain

'##################################################
gameover:
  PLAY se$(6)
  GOSUB waiter2
  GOSUB keywait: CLS : LOCATE 8, 1:
  LOCATE 1, 1:
  '     PRINT "                                   ":
        PRINT "          ----GAME OVER----        ":
  PRINT : PRINT : PRINT : PRINT :
  PRINT USING "            Round ######            "; round:
  PRINT
  PRINT USING "     Your Score is ######            "; score:
  PRINT : COLOR 13
        PRINT "       "; name$
  PRINT USING "       High-score is ######           "; highscore:
  PRINT
  PRINT USING "       High-round is ######           "; highround
  COLOR 15
  IF highscore < score THEN
    highscore = score: highround = round
    PRINT : PRINT : PRINT :
    COLOR 14: PRINT " You are New Top Player! Good play.": COLOR 15
    PLAY se$(0)
    INPUT "What's Your Name?", name$
    IF name$ = "" THEN name$ = "Secret"
    OPEN "pplane.dat" FOR OUTPUT AS #1
    PRINT #1, highscore
    PRINT #1, highround
    PRINT #1, name$; "("; DATE$; ")"
    CLOSE
    END IF
  GOSUB keywait
  GOTO title:


'##################################################
plane:
 'player Key check
  backangle = 0
  IF INP(&H60) = 57 THEN
    backangle = 2
    IF flyflag < 3 THEN v = v + 1 - flyflag * .2: flyflag = flyflag + 1
  ELSE flyflag = 0
  END IF

  'angle culc---------------
  w = -2 * COS(angle * pi) + backangle * v * .3
  IF v = 0 THEN IF angle > 16 THEN w = w + 1 ELSE w = w - 1
  angle = angle + w

  '-------place culc---------------
  vx = v * COS(pi * angle)
  vy = v * SIN(pi * angle) + gravity
  vy = -vy
  v = v + gravity * SIN(pi * angle)
  px = px + vx: py = py + vy

  IF v < 0 THEN v = 0
  IF v > 6 THEN v = 6
  IF px < 8 THEN px = 8
  IF px > 324 THEN px = 8: GOSUB goal
  IF py < 0 THEN py = 0
  IF py > 194 THEN py = 194
  angle = angle AND 63
  'for debug
  'LOCATE 2, 3: PRINT USING "### ### ### ###"; px; py; vy; angle
 RETURN


plane.draw:
  LINE (pre.px, pre.py)-(pre.px + pre.xl, pre.py + pre.yl), 0
  LINE (pre.px, pre.py)-(pre.px + pre.x2, pre.py + pre.y2), 0
  LINE (pre.px + pre.xl, pre.py + pre.yl)-(pre.px + pre.x2, pre.py + pre.y2), 0
  xl = COS(pi * angle) * 8
  yl = -SIN(pi * angle) * 8
  x2 = COS(pi * (angle - backangle * 4)) * -5
```

★影前線が北上してくる。お腹守らなくちゃッ！（愛知県一宮市・幻18歳）……【**編**：いやー、今日も暑かったですねー。そろそろ仕事切り上げましょうか。影前線が発生しないうちに。**編集長**：よし、一杯やってこうか？**影**：私もー。**編集長**：サイフ守らなくては。】

```
    y2 = -SIN(pi * (angle - backangle * 4)) * -5

    'LINE (tmpx * 8, tmpy * 8)-(tmpx * 8 + 7, tmpy * 8 + 7), 13, B

    LINE (px, py)-(px + x1, py + y1), 15
    LINE (px, py)-(px + x2, py + y2), 15
    LINE (px + x1, py + y1)-(px + x2, py + y2), 15
    pre.px = px: pre.py = py
    pre.x1 = x1: pre.x2 = x2
    pre.y1 = y1: pre.y2 = y2
RETURN

'# Block's Proc #################################
block.init: '--------------------------------------
  FOR i = 0 TO 39
    FOR j = 0 TO 22
      bmap(i, j) = 0
      IF j = 0 OR j = 22 THEN bmap(i, j) = 8
    NEXT
  NEXT
  FOR i = 0 TO 32:
    bx(i) = 0: by(i) = 0: bt(i) = 0: pre.bx(i) = 0: pre.by(i) = 0:
  NEXT
  bnum = 0:
RETURN

block.proc: '--------------------------------------
  IF RND * 100 < (round * 10 + time AND 300) THEN GOSUB block.generate
  a = 0: b = bnum
  FOR i = 1 TO b
    IF bt(i) > 0 THEN
      bt(i) = bt(i) - 1
      IF bt(i) < 3 THEN ccc = 4 ELSE ccc = 6
      LINE (bx(i) * 8, by(i) * 8)-(bx(i) * 8 + 7, by(i) * 8 + 7), ccc, BF
      bmap(bx(i), by(i)) = 6:
      a = a + 1
      bx(a) = bx(i): by(a) = by(i): bt(a) = bt(i)
    ELSE
      IF bmap(bx(i), by(i) + 1) = 0 THEN
        bmap(bx(i), by(i)) = 0:
        LINE (bx(i) * 8, by(i) * 8)-(bx(i) * 8 + 7, by(i) * 8 + 7), 0, BF
        by(i) = by(i) + 1
        bmap(bx(i), by(i)) = 6:
        LINE (bx(i) * 8, by(i) * 8)-(bx(i) * 8 + 7, by(i) * 8 + 7), 6, BF
        a = a + 1
        bx(a) = bx(i): by(a) = by(i): bt(a) = bt(i)
      ELSE bnum = bnum - 1: PLAY se$(2)
      END IF
    END IF
  NEXT
RETURN

block.generate: '----------------------------------
  rrrr = round + 5: IF rrrr > 25 THEN rrrr = 25
  IF bnum < rrrr + 5 THEN
    bnum = bnum + 1: rrrr = (round MOD 5) * 10
    bx(bnum) = RND * 34 + 5: by(bnum) = 1: bt(bnum) = -RND * rrrr + 50
  END IF
RETURN

block.display: '-----------------------------------
  FOR i = 0 TO 39
    FOR j = 0 TO 22
      LINE (i * 8, j * 8)-(i * 8 + 7, j * 8 + 7), bmap(i, j), BF
    NEXT
  NEXT
RETURN

block.del: '---------------------------------------
  blockline = 20 - round / 7
  IF blockline < 7 THEN blockline = 7
  FOR j = blockline TO 1 STEP -1
    FOR i = 0 TO 39
      IF bmap(i, j) <> 0 THEN
        score = score + 1
        bmap(i, j) = 0
        LINE (i * 8, j * 8)-(i * 8 + 7, j * 8 + 7), 7, BF
      END IF

    NEXT
  NEXT
  bnum = 0
RETURN

'################################################
goal: '--------------------------------------
  PLAY se$(4)
  round = round + 1
  score = score + 30 + round * 10
  IF (round MOD 7) = 0 THEN GOSUB block.del: GOSUB waiter2: GOSUB block.display
  IF round MOD 10 = 5 THEN gamewait = gamewait - 1
  IF round MOD 10 = 9 THEN gamewait = gamewait + 1
  IF round = 30 THEN gamewait = gamewait - 1

RETURN

keywait: '--------------------------------------
  LOCATE 24, 1: COLOR 4:
  PRINT "            PUSH SPACE KEY!!                ";
  WHILE INP(&H60) <> 57: a$ = INKEY$: WEND
  WHILE INP(&H60) = 57: a$ = INKEY$: WEND
  LOCATE 24, 1:
  PRINT "                                            ";
  PLAY se$(1): COLOR 15
RETURN


waiter: '--------------------------------------
  FOR wwww = 0 TO gamewait
    WHILE (INP(&H3DA) AND 8) <> 0: WEND
    WHILE (INP(&H3DA) AND 8) = 0: WEND
  NEXT
  WHILE INKEY$ <> "": WEND
RETURN

waiter2: '--------------------------------------
  FOR zzzz = 0 TO 10: GOSUB waiter: NEXT
RETURN

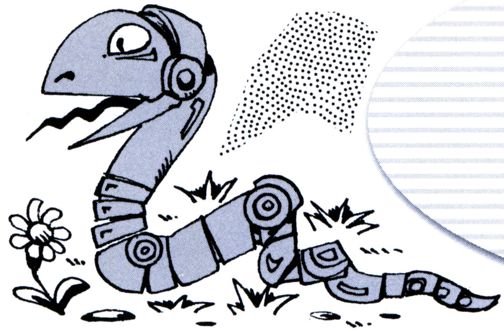
makedatafile:
CLS
PRINT "Make Data File."
OPEN "pplane.dat" FOR OUTPUT AS #1
PRINT #1, 0
PRINT #1, 0
PRINT #1, "Top score("; DATE$; ")"
CLOSE
PRINT "Ok, Success."
END
```

★7月号の表紙から特選投稿プログラムコーナーのタイトルが強調されていて、とてもよかったです。ベーマガのプログラムに対する意気込みが感じられてうれしいです。(島根県松江市・NSR250R＋横浜ナンバー37歳)……【編：プログラムの楽しさを広げたいですね。】

ワンダースワン(WonderWitch)用

なつかしのスタンダードゲーム

スネーク



中野　真良（25歳）

内容

スネークを操作して障害物をさけ，えさを食べていくゲームです。ステージは7面あり，進むごとに難易度が上がっていきます。

操作方法

スネーク「●」を操作して，「○」を食べるとパワーアップして体の大きさとスピードがUPします。「◎」でからだが小さくなり「⑪」でスピードが遅くなります。「□」に触れたり，自分の体に触れるとゲーム・オーバーです。

操作は，X1～X4キーでそれぞれ上右下左に対応していて，スタートボタンでゲーム中断します。

注意点

OSのバージョンが1.0.0で開発したので，最新OSは？です。あわててOS1.1.1でデバックしましたが不具合はないと思います。

最後に

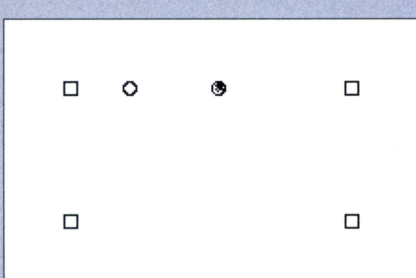
スプライト知識がいまいちなので恥ずかしいプログラムを書いちゃってごめんなさい。このゲームは，中学のころBASICで遊んだ記憶があって，この前パチンコをしていて受け皿の玉と玉がくっついたとき，バックフラッシュのごとく思い出しました。ゲームウォッチに親しんだ私としてはワンダーウィッチでしょってことです。

CHECKER FLAG

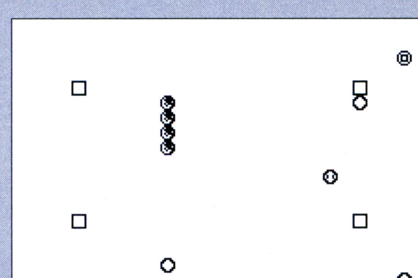
影：説明文にもあるように昔懐かしのゲーム。WonderWitchのサンプルにもスネーク・ゲームがあるけど，それとはまた別のタイプだ。

つぐ美：スプ ライトによるキャラクターとテキスト表示のキャラクターをうまく組み合わせて利用しているわね。

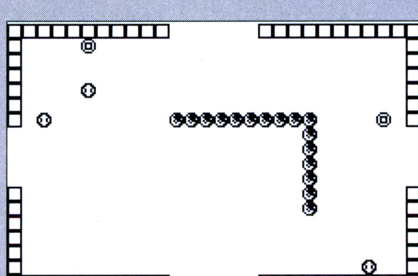
Dr.D：リスト面での不備として，スタートボタンによる中断処理にバグがあるゾ。スタートを押してゲーム状態のフラグ(GameFlg)をストップ状態にしたあと，それを元に戻す記述が書かれていないためじゃ。スタートボタンを押したときに，GameFlgの値に合わせた処理をする必要があるゾ。



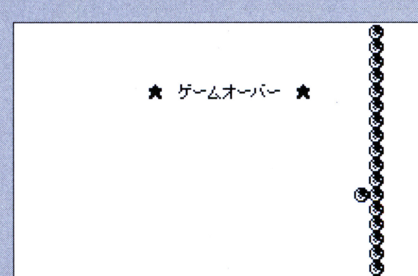
《写真1》スタンダードなゲーム。遊んだことある？



《写真2》"○"をとってスネークの体を大きくしていく



《写真3》ステージやアイテムも複数用意



《写真4》自分の体にぶつかるとゲーム・オーバー

★ついに僕のMIDI音源が壊れた。新たに音源内蔵シーケンサを買った。あぁ～，5万円がぁぁぁぁ…。（福島県郡山市・**近内沖尋18歳**）……【**影**：うらやましい。私の場合，サングラスを買い換えたい。**つぐ美**：その前に靴と背広を新調したほうがいいと思いますよ。】

《リスト１》snake.cのプログラム・リスト

```
/* snake */

#include <stdio.h>
#include <stdlib.h>
#include <sys/bios.h>

#include "snake.h"

#define SCR_X_MAX 28
#define SCR_Y_MAX 18

#define MAX_SPRITE 128

#define SPR_CHAR_NO_PLY  0
#define CHAR_NO_ATM1 1
#define CHAR_NO_ATM2 2
#define CHAR_NO_ATM3 3

#define SPR_PALETTE_NO  12

#define MAX_CHAR_NO  (snake_width * snake_height)
#define MAX_CHAR_NO2 (snake2_width * snake2_height)

#define PLY_SNAKE 0  /* 0..31*/
#define NON  0
#define KABE 1
#define SNK  2
#define ATM1 3  /* ◎ スネーク半分 */
#define ATM2 4  /* ○ スネーク２倍 */
#define ATM3 5  /* ⑪ スピード半分 */

#define TITLE_GAME 0
#define START_GAME 1
#define STOP_GAME  2
#define GAME_OVER  3
#define GAME_CLEAR 4

#define HIT 0
#define SHT 1

#define NEXTSTAGE 10
#define ATMPIC_SUU 5
#define ATMCNT_SUU 7

int Wait;
int x[32+1],y[32+1];
int map[SCR_X_MAX][SCR_Y_MAX];
int AtmX[ATMPIC_SUU],AtmY[ATMPIC_SUU];
int snake_suu;
int Atmchk;

void settei(void);
void MapSet( int );
int atari(void);
int A_atari(void);
void AtmXYReset( int, int );


void main(void)
{
    int i,j,x_ido,y_ido,key;
    int GameFlg;
    int stage;
    int AtemCnt,AtemSuu,AtmKind,AtmGetSuu,Atm_x,Atm_y;

    text_screen_init();
    sprite_set_range(0,MAX_SPRITE - 1);
```

```
display_control(DCM_SCR1 | DCM_SCR2 | DCM_SPR);

srand(rtc_get_datetime(RTC_SEC));  /* 乱数を現在の秒で初期化 */

settei();

/* ステージ設定 */
stage = 1;

GameFlg = TITLE_GAME;

/* メインループ */
while(1){

    /* 初期設定 */
    if( GameFlg == TITLE_GAME){
        /* スネーク初期値 */
        x[0] = 23;
        y[0] =  3;
        x_ido = -1;
        y_ido = 0;
        snake_suu = 1;
        Wait = 30;
        AtemCnt = 1;
        AtemSuu = 0;
        AtmGetSuu = 0;

        MapSet( stage );

        GameFlg = START_GAME;
    }

    for(i=0; i<=Wait; i++){  /* ウエイト */

        /* キー入力判定 */
        key = key_hit_check_with_repeat();
        if( key != 0 ){
            if(key & KEY_LEFT1){    /* 左ボタン */
                x_ido = -1;
                y_ido =  0;
            }
            if(key & KEY_RIGHT1){   /* 右ボタン */
                x_ido = 1;
                y_ido = 0;
            }
            if(key & KEY_UP1){      /* 上ボタン */
                y_ido = -1;
                x_ido =  0;
            }
            if(key & KEY_DOWN1){    /* 下ボタン */
                y_ido = 1;
                x_ido = 0;
            }
            if(key & KEY_START){    /* スタートボタン */
                GameFlg = STOP_GAME;
                break;
            }
        }
    }

    /* ゲーム中断 */
    if( GameFlg == STOP_GAME ){
        break;
    }

    /* 子スネーク座標設定 */
    for(i=snake_suu; i>0; i--){
        x[i] = x[i-1];
```

★影さんの似顔絵コンテスト開催。影さん自身が気に入った似顔絵を描いてくれた人の中から抽選で1名に、影さんを1日お貸しします。（宮城県仙台市・light13歳）……【影：おいおい、勝手に私を賞品にしないでおくれよ。3食たらふく食べさせてくれるならOK！】

★「VBゲーム作成術」があるのに、なぜ「影さんのパソコン破壊術」がないのですか？（神奈川県藤沢市・たっくん16歳）……【影：それは誰も望んでないからなんだよ。しかも、特別なテクニックを使ってるわけじゃないのに、パソコンが勝手に壊れちゃうんだよ。】

```
        y[i] = y[i-1];
    }

    /* すり抜け座標設定 */
    /* X軸 */
    if( ( x[0] + x_ido) < 0 ){
        x[0] = SCR_X_MAX;
    }
    if( ( x[0] + x_ido) >= SCR_X_MAX ){
        x[0] = -1;
    }

    /* Y軸 */
    if( ( y[0] + y_ido) < 0 ){
        y[0] = SCR_Y_MAX;
    }
    if( ( y[0] + y_ido) >= SCR_Y_MAX ){
        y[0] = -1;
    }

    /* 親スネーク座標設定 */
    x[0] = x[0] + x_ido;
    y[0] = y[0] + y_ido;

    /* スネーク表示処理 */
    for(i=0; i<snake_suu; i++){
        sprite_set_location(PLY_SNAKE+i, (8 * x[i]), (8 * y[i]));
    }

    /* アイテム処理 */
    AtemCnt++;
    if( AtemCnt >= ATMCNT_SUU ){
        AtemCnt = 0;
        /* 古いアイテムを消去 */
        if( AtemSuu >= ATMPIC_SUU ){
            screen_fill_char(SCREEN2, AtmX[0], AtmY[0], 1, 1, 7);
            map[AtmX[0]][AtmY[0]] = NON;
            for( i=0; i<AtemSuu-1; i++ ){
                AtmX[i] = AtmX[i+1];
                AtmY[i] = AtmY[i+1];
            }
            AtemSuu = 4;
        }
        AtemSuu++;
        AtmKind = ( ( rand() % 4 ) + 3 );
        if( AtmKind >= 6 ){  /* 「○」だけ出現確立２倍 */
            AtmKind = ATM2;
        }
        for(i=0; i<10; i++){
            Atm_x = rand() % SCR_X_MAX;
            Atm_y = rand() % SCR_Y_MAX;
            if( map[Atm_x][Atm_y] == NON ){
                map[Atm_x][Atm_y] = AtmKind;
                AtmX[AtemSuu-1] = Atm_x;
                AtmY[AtemSuu-1] = Atm_y;
                screen_fill_char(SCREEN2, Atm_x, Atm_y, 1, 1, (AtmKind - 2) );
                break;
            }
        }
    }

    /* あたり判定 */
    if( atari() == HIT ){
        GameFlg = GAME_OVER; /* ゲームオーバー */
        break;
    }

    /* アイテムあたり判定 */
```

```
        AtmKind = A_atari();
        switch( AtmKind ){
            case ATM1:
                if( snake_suu > 1 ){
                    j = snake_suu / 2;
                    for(i=snake_suu; i>j; i--){
                        sprite_set_location(PLY_SNAKE+snake_suu-1, -8, -8);
                        snake_suu = snake_suu - 1;
                    }
                }
                AtmXYReset( AtemSuu, Atmchk );
                AtemSuu--;
                screen_fill_char(SCREEN2, x[Atmchk], y[Atmchk], 1, 1, 7);
                break;

            case ATM2:
                if( snake_suu < 32 ){
                    j = snake_suu * 2;
                    for(i=snake_suu; i<j; i++){
                        x[snake_suu] = x[snake_suu - 1];
                        y[snake_suu] = y[snake_suu - 1];
                        sprite_set_location(PLY_SNAKE+snake_suu, (8 * x[snake_suu]), (8 * y
[snake_suu]));
                        snake_suu = snake_suu + 1;
                    }
                    if( Wait > 5 ){
                        Wait = Wait - 5;
                    }
                }
                AtmXYReset( AtemSuu, Atmchk );
                AtmGetSuu++;
                AtemSuu--;
                screen_fill_char(SCREEN2, x[Atmchk], y[Atmchk], 1, 1, 7);
                break;

            case ATM3:
                if( Wait < 30 ){
                    Wait = Wait + 5;
                }
                AtmXYReset( AtemSuu, Atmchk );
                AtemSuu--;
                screen_fill_char(SCREEN2, x[Atmchk], y[Atmchk], 1, 1, 7);
                break;
        }

        if( AtmGetSuu >= NEXTSTAGE ){
            text_screen_init();
            text_put_string(9, 4, "★ ＮＥＸＴステージ ★");
            stage++;
            key_wait();
            GameFlg = TITLE_GAME;
            if( stage >= 8 ){
                /* クリアー */
                GameFlg = GAME_CLEAR;
                break;
            }
        }
    }

    text_screen_init();
    switch( GameFlg ){
        case STOP_GAME:
            text_put_string(9, 4, "★ ゲーム中断 ★");
            break;

        case GAME_OVER:
            text_put_string(9, 4, "★ ゲームオーバー ★");
            break;
```

```
        case GAME_CLEAR:
            text_put_string(9, 4, "★ ゲームクリア ★");
            break;
    }
    key_wait();
}

void settei()
{

    int i;

    font_set_colordata(0, MAX_CHAR_NO,  bmp_snake);
    font_set_colordata(4, MAX_CHAR_NO2, bmp_snake2);

    palette_set_color(SPR_PALETTE_NO, 0x7300);

    /* スプライト設定 */
    for(i=0; i<32; i++){
        sprite_set_char_location(PLY_SNAKE+i,   CFM_SPR_UPPER | ((SPR_PALETTE_NO - 8) << 9)
 | SPR_CHAR_NO_PLY, -8, -8);
    }
}

void MapSet( int stage )
{
    int i,j;

    screen_fill_char(SCREEN2,  0,   0,  32, 32, 7);

    for(i=0; i<32; i++){
        sprite_set_location(PLY_SNAKE+i, -8, -8);
    }

    /* map初期化 */
    for(i=0; i<SCR_X_MAX; i++){
        for(j=0; j<SCR_Y_MAX; j++){
            map[i][j] = NON;
        }
    }

    switch( stage ){
        /* ステージ1 */
        case 1:
            for(j=0; j<SCR_Y_MAX; j++){
                for(i=0; i<SCR_X_MAX; i++){
                    if( stage_a[ i + ( j * SCR_X_MAX ) ] == 1 ){
                        screen_fill_char(SCREEN2, i, j, 1, 1, 5);
                        map[i][j] = KABE;
                    }
                }
            }
            break;

        /* ステージ2 */
        case 2:
            for(j=0; j<SCR_Y_MAX; j++){
                for(i=0; i<SCR_X_MAX; i++){
                    if( stage_b[ i + ( j * SCR_X_MAX ) ] == 1 ){
                        screen_fill_char(SCREEN2, i, j, 1, 1, 5);
                        map[i][j] = KABE;
                    }
                }
            }
            break;

        /* ステージ3 */
        case 3:
            for(j=0; j<SCR_Y_MAX; j++){
                for(i=0; i<SCR_X_MAX; i++){
                    if( stage_a[ i + ( j * SCR_X_MAX ) ] == 1 || stage_b[ i + ( j * SCR_X_M
AX ) ] == 1 ){
                        screen_fill_char(SCREEN2, i, j, 1, 1, 5);
                        map[i][j] = KABE;
                    }
                }
            }
            break;

        /* ステージ4 */
        case 4:
            for(j=0; j<SCR_Y_MAX; j++){
                for(i=0; i<SCR_X_MAX; i++){
                    if( stage_c[ i + ( j * SCR_X_MAX ) ] == 1 ){
                        screen_fill_char(SCREEN2, i, j, 1, 1, 5);
                        map[i][j] = KABE;
                    }
                }
            }
            break;

        /* ステージ5 */
        case 5:
            for(j=0; j<SCR_Y_MAX; j++){
                for(i=0; i<SCR_X_MAX; i++){
                    if( stage_a[ i + ( j * SCR_X_MAX ) ] == 1 || stage_c[ i + ( j * SCR_X_M
AX ) ] == 1 ){
                        screen_fill_char(SCREEN2, i, j, 1, 1, 5);
                        map[i][j] = KABE;
                    }
                }
            }
            break;

        /* ステージ6 */
        case 6:
            for(j=0; j<SCR_Y_MAX; j++){
                for(i=0; i<SCR_X_MAX; i++){
                    if( stage_b[ i + ( j * SCR_X_MAX ) ] == 1 || stage_c[ i + ( j * SCR_X_M
AX ) ] == 1 ){
                        screen_fill_char(SCREEN2, i, j, 1, 1, 5);
                        map[i][j] = KABE;
                    }
                }
            }
            break;

        /* ステージ7 */
        case 7:
            for(j=0; j<SCR_Y_MAX; j++){
                for(i=0; i<SCR_X_MAX; i++){
                    if( stage_a[ i + ( j * SCR_X_MAX ) ] == 1 || stage_b[ i + ( j * SCR_X_M
AX ) ] == 1 || stage_c[ i + ( j * SCR_X_MAX ) ] == 1 ){
                        screen_fill_char(SCREEN2, i, j, 1, 1, 5);
                        map[i][j] = KABE;
                    }
                }
            }
            break;
    }
    sprite_set_location(PLY_SNAKE, (23 * 8), (3 * 8) );
    key_wait();
}

int atari(void)
{
```

★パソコンショップのインターネット体験コーナー（占拠中）から、はじめてのインターネット投稿です。（東京都練馬区・ルストフレム14歳）……【編：えらい！今はどこにいてもインターネットに接続できる環境があるからね。プログラミングもショップのPCで！】

```
{
    int i;

    /* 障害物判定 */
    for(i=0; i<snake_suu; i++){
        if( map[ x[i] ][ y[i] ] == KABE ){
            return( HIT );
        }
    }

    /* 自機判定 */
    if( snake_suu >= 2 ){
        for(i=1; i<snake_suu; i++){
            if( (x[0] == x[i]) && (y[0] == y[i]) ){
                return( HIT );
            }
        }
    }
    return( SHT );
}

int A_atari(void)
{
    for(Atmchk=0; Atmchk<snake_suu; Atmchk++){
        switch( map[ x[Atmchk] ][ y[Atmchk] ] ){
            case ATM1:
                map[x[Atmchk]][y[Atmchk]] = NON;
                return( ATM1 );
            case ATM2:
                map[x[Atmchk]][y[Atmchk]] = NON;
                return( ATM2 );
            case ATM3:
                map[x[Atmchk]][y[Atmchk]] = NON;
                return( ATM3 );
        }
    }
    return( NON );
}

void AtmXYReset( int AtemSuu, int Atmchk )
{
    int i,j;

    for( i=0; i<AtemSuu; i++ ){
        if( AtmX[i] == x[Atmchk] ){
            if( AtmY[i] == y[Atmchk] ){
                for( j=i; j<AtemSuu-1; j++ ){
                    AtmX[j] = AtmX[j+1];
                    AtmY[j] = AtmY[j+1];
                }
                break;
            }
        }
    }
}
```

《リスト 2》snake.hのプログラム・リスト

```
#define snake_width  2
#define snake_height 2

#define snake2_width  2
#define snake2_height 2

unsigned short bmp_snake[] = {
    0x3C3C, 0x5E7E, 0xBFFF, 0x9DFF, 0xAFFF, 0x93FF, 0x467E, 0x3C3C,
    0x3C3C, 0x4242, 0xBDBD, 0xA5A5, 0xA5A5, 0xBDBD, 0x4242, 0x3C3C,
    0x3C3C, 0x6666, 0xC3C3, 0x8181, 0x8181, 0xC3C3, 0x6666, 0x3C3C,
    0x3C3C, 0x6666, 0xC3C3, 0xA5A5, 0xA5A5, 0xC3C3, 0x6666, 0x3C3C,
};

unsigned short bmp_snake2[] = {
        0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
        0xFFFF, 0x8181, 0x8181, 0x8181, 0x8181, 0x8181, 0x8181, 0xFFFF,
        0xFFFF, 0xC3C3, 0xA5A5, 0x9999, 0x9999, 0xA5A5, 0xC3C3, 0xFFFF,
        0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,

};

int stage_a[] = {
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
};

int stage_b[] = {
    1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,
    1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
    1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
    1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
    1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
    1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
    1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
    1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
    1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
    1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
    1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
    1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
    1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1
};

int stage_c[] = {
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,0,0,0,0,0,0,
    0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0,0,0,0,0,0,
    0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
};
```

《リスト 3》snake.cfのプログラム・リスト

```
name: snake
info: snake
mode: 7
source: snake.bin
output: snake.fx
```



CMM試聴室

毎月みなさんの曲を紹介するコーナーです。募集要項をご覧のうえ，どしどしご応募ください。ここで紹介しているデータは，CMMのWebSite（http://www.basicmagazine.net/cmm/）に収録されています。曲への意見，アドバイスなども大歓迎します。

スプリング・ラブ ………自作

投稿者：安達学人さん（20歳）

E-Mail：dm7_g7@hotmail.com
制作音源 SC-8820

明るい雰囲気の中にマイナー感を出しました

ピアノ・トリオによるバラードです。……といってもジャズではないですけど。タイトル通り，春の穏やかで明るい雰囲気の中にちょっとしたマイナー感をだそうとしました。一部でごく小さい音でストリングスが入っている以外，他の音を使いたくなる気持ちを抑えて自分なりにシンプルにまとめたつもりです。サビの前半は音を大きめにして盛り上げ，後半は対照的に曲の中で一番静かに，そして，サビの2回目は同じメロディでリズムを変えて，はずんだ感じにしました。効果のほどはどうでしょうか？　あと，僕はいわゆるエフェクトなどの音作りが苦手なんですけど，何か良いアドバイスがありましたら教えてください。

編集部から

BRUSHのドラムキットがやわらかい雰囲気をだしている，ゆったりとした曲です。ピアノのフレージングはところどころちょっとぎこちない感じがしますけど，全体的にはおもしろい響きになっていていいと思います。最後のほうで，イントロと同じメロディに戻るところのつなぎかたが不自然なので，ここにはもう1場面くらいはさんであげるといいでしょう。エフェクトについてですが，空間系（リバーブなど）とそれ以外（ディストーションなど）で変わってきますが，空間系の場合は，楽器を演奏する場所と，楽器の位置を頭に思い浮かべながら，パンやリバーブの量を調整して，空間の広さを変える場合はリバーブのキャラクター（種類）を変えるというのはどうでしょうか。

Moon Rider ………自作

投稿者：NAOKIさん（30歳）

E-Mail：AmaMetal@aol.com
制作音源 SC-88Pro

2年ぶりの投稿です

こんにちは。NAOKIと申します。約2年ぶりの投稿をさせていただきます。しばらくの間仕事やらなんやらで，コンピュータ・ミュージックの世界から遠く離れていましたが，ようやくそれも一段落し，もう1回DTMの世界に足を踏み入れてみようと思った次第です。あれほどお世話になっていたCakewalkの使いかたもすっかり忘れ，どうしたものかと途方にくれましたが，地道な譜面入力によりなんとか完成させることができました。かなり幼稚なデータとは思いますが，曲自体のクオリティは，昔のものより良いのではないかと思っています。曲のジャンル的には前と変わらず，ソリッドなロックを目指したつもりです。またいろいろ教えていただきたいです。あと，いろんな感想やアドバイス待ってます。

編集部から

スピード感のあるロック調のオリジナル曲で，すっきりとした構成で聴きやすい曲です。全体のミックスもうまくまとめられていると思います。途中でメロディの楽器が変わったり，エンディングでフェードアウトしながら終わる部分など，工夫されていておもしろいです。

曲的に気になったのは，メロディの音階が下がる部分です。とくに，コードの変わり目でメロディの音階が下がる部分で，ちょっと大きく下がりすぎかな，と聴こえる部分が多いです。このような，流れるような感じの曲だととくに，メロディのつながりかたが重要なポイントになってきますので，コードの構成音であると同時に，スムーズなメロディの連結について工夫してみてください。また，メロディがブラスに変わる部分ですが，もう少しアタックの速いブラスのほうがメリハリが出ると思います。

投稿上の注意

投稿は，自分で入力したオリジナル曲で，未発表のものに限ります。市販のMIDIデータ曲集や，パソコン通信ネット，インターネットのWebSiteにアップされているものを改造しただけのデータはお断りします。ただし，パソコン通信，インターネットのWebSiteについては，投稿者本人が公開しているものはOKです。

掲載された曲に関しては，規定の原稿料をお支払いします（掲載誌発売後1か月程度かかります）。曲データは，CMMのWebSite（http://www.basicmagazine.net/cmm/）にも，コメントとともにアップロードされます。

CMMのWebSite上，もしくはE-Mailによる投稿も受け付けています。

●あて先・問い合わせ先
〒141-8715　東京都品川区東五反田1-11-15
電波新聞社　マイコンBASICマガジン編集部
「CMM試聴室」係
☎ 03-3445-6111
E-Mail cmm@basicmagazine.net

マイコンBASICマガジン

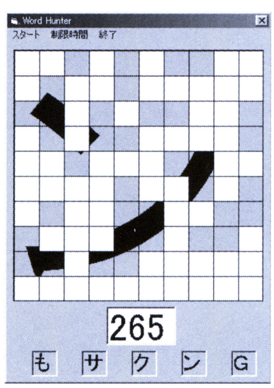
明日のスター☆プログラマー

今月のベスト・プログラマー

編：リストがコンパクトにまとまっている作品が目立った今月号。チェックしている段階では，VB用の『Word Hunter』に対する評価が高かったのですが，あらためて，みんなの意見を聞いてみましょう。

影：全体を見直してみても，やっぱ『Word Hunter』が一歩抜きんでている感じだなぁ。ディレクトリ構造をゲームに使用するというアイデアを実現するための，プログラム側の対処もしっかりしているし，リスト面から見てもレベルが高いと言えるんじゃないかな？

つぐ美：言語面で注目してみると，今月がんばっているのはQBasicね。『Paper Plane Forever　紙飛行機よ永遠なれ』，『Mouse対応版SHORTEN』ともに遊びやすく作られているので，両作品ともチェックしていて楽しめました。



▲VB用『Word Hunter』

Dr.D：ただ，アイデア面ではさびしい作品が多いと言わざるを得ないじゃろう。編くんが言っておるように，リストをコンパクトにまとめている作品が多いのは評価できるが，ゲーム内容自体は，どこかで一度は見たことがあるような作品ばかりじゃ。ゲーム・システム全体において独自のアイデアを出すのは難しいじゃろうが，ある1か所にでも，自分独自のアイデアを入れる，というこだわりがほしいところじゃ。そういう面で『Word Hunter』は，パズルゲームにディレクトリ構造を使うという，こだわりがある作品と言えるじゃろう。

つぐ美：せっかくプログラムでオリジナルな作品を作るのなら，自分ならではという工夫を入れたほうがおもしろいし，作ったあとに愛着みたいなものが芽生えますものね。

Dr.D：そういうことじゃ。「他人とは一味ちがったゲームを作るぜ！」といった意気込みを持ってプログラムを作成してみてはどうかな？

編：Dr.のありがたい助言をもらったところで，ベストの発表といきましょう。今月はチェック段階の前評判どおり『Word Hunter』がふさわしいようですね。

Dr.D：みんなの評価も高いし，妥当な選考だと思うぞ。

つぐ美：というわけで，今月のベスト・プログラマーは，VB用『Word Hunter』の吉田真人さんが受賞です。おめでとうございます。

編：さて，先月号と同じようにベストに選ばれた作品は「Dr.Dのベスト・プログラム研究室」のページでDr.Dがよりくわしく解説しています。ぜひそちらもチェックしてね。

影さんのマジック・ボックス？
ワンテク&アイデア募集コーナー

このコーナーは，みなさんのプログラミングに関するワンポイント・テクニックや，アイデアなどを紹介していこうというもの。投稿する際は，機種や動作環境などもしっかり明記してください。

あて先：〒141-8715　東京都品川区東五反田1-11-15 電波新聞社
マイコンBASICマガジン編集部
「影さんのマジック・ボックス」係
e-mail
Webmas@basicmagazine.net
件名に，影さんのマジック・ボックス係と記入してください。

勝手に関連付けを変えさせない!!

関連付け㊙ノウハウ

Windows95以降

他人が勝手にエクスプローラの[表示−ｵﾌﾟｼｮﾝ]コマンドを使ったり、既成アプリケーションの自動関連付けによって設定が変更されて困惑する方は多いと思います。そこで、レジストリの項目を駆使してみましょう!!

レジストリの[ﾏｲ ｺﾝﾋﾟｭｰﾀ¥HKEY_CLASSES_ROOT]以下のキーが、それぞれのオブジェクトにおける処理方法を設定していますが、バイナリの値[EditFlags]に[01000000]を代入すれば、以後、エクスプローラの[表示−ｵﾌﾟｼｮﾝ−ﾌｧｲﾙﾀｲﾌﾟ]の一覧には表示されなくなります。いたずらに変更されると困るような種類のファイルに関しては、この方法で対処すると良いと思います。

しかし、既成アプリケーションソフトが、勝手に設定を変えてしまうのは、この方法では対応しきれません。そこで、絶対に変えてもらいたくない設定に関しては、まるごとその項目をレジストリの項目ファイル(REGファイル)として保管してしまいましょう。そして、スタートアップ時の最後尾で起動するようにします。この方法は、一旦[ｽﾀｰﾄｱｯﾌﾟ]フォルダにあるファイルをすべて取り出し、再び放り込み、更にその後にレジストリの項目ファイルを入れればOK!!

システムに関するオブジェクトには絶対に使用しないで下さい。スタートアップ時はマシンが不安定になりやすく、万が一の場合、致命的な故障を引き起こします。



▲加藤義啓&サボタン

投稿ありがとう

はりまお／ていこく／#Shin.Y／西野紀之／よしとも／マギヲ／赤羊旅団／山崎啓吾／RyuZyu／小澤修一／橘　一馬／吉田真人／M.Hayashi／梶　夏樹／宮尾　歩／tayosan／小薗直哉／山屋　健／畑の木／中野真良／金井義男／Mr.親父／平川貞春／やまたろう／阿部真治／横岩通彦／太田　栄／藤本　豊／マイクロ・ウッド／吉井智衛／佐々木一彦／吉村俊一／多橋幸広／ATTO／木沢謙一
※全員紹介できなくてごめんなさい

投稿プログラム・コーナー

Programmer's Room

今月の総評

編：全体的に投稿が少な目だった今月号。採用作品にも，もう一つ抜き出た作品が少なかったように思います。来月以降の投稿に期待したいですね。また今の時期，学生のみなさんは夏休みの真っ最中。この時間を利用して，プログラム投稿にチャレンジしてみてはどうでしょうか？

●ゲーム・バランスとは？

『らむしふと』の作者，赤羊旅団さんの説明文内にあった質問。CHECKER FLAGでも少し触れましたが，これについてもう少し補足していきましょう。

CHECKER FLAGでは，ゲーム・バランス＝遊びやすさ，と言いましたが，この遊びやすさとは何でしょうか？

遊びやすさに直接かかわってくるのは難易度です。投稿にありがちなのが，難易度が極端に高いゲーム。これは制作者しかチェックしていないのが大方の原因ですね。

つぎにかかわってくるのが，ゲーム・スタート時の内容だと思っています。いきなりゲームがスタートしたり，はたまたパズルゲームなどでは，1面からルールの裏をかくような思考が要求される作品があります。こういった作品は，いくら完成度が高くても，あまり印象がよくありません。

当然のことですが，プログラムをチェックする私たちは，投稿される作品を初めてプレイします。いくら説明文を読んでいるとはいえ，やはりプレイしてみて実感できる操作やルールというものが存在します。

だからこそ，ゲームの最初に基本的な操作やルールを把握させる場面があると遊びやすいと感じますし，そういった作品はプレイしていて楽しいものです。

いろいろ述べてきましたが，ゲーム・バランスを考えるにあたって大切なのは，“遊ぶ人のことを考えているか？”という点ではないでしょうか？せっかく作った作品なのですから，その作品をプレイする人が楽しめる気配りがほしいものですね。

一口CHECKER FLAG

●Visual Basic用『ナンバーベースボール』／梶　夏樹（19歳）

つぐ美：三つの数字を当てる基本的なゲーム。アイテムなどといった工夫はあるのですが，フォームの構成がスッキリしていないのでチョット遊びにくい感じね。あと，投稿にはfrmファイルも一緒に送ってね。また，個人的な意見ですが，指定した三つの数字がすべて該当しない(空振り）のが，減点対象になるのはいただけないわ～。あれって逆に当たりでしょ？

●Visual Basic用『モグラ叩き』／西野紀之（13歳）

影：まず，リスト面での注意。パスの設定が，個人のパソコンに依存する形式のため，そのままでは実行できなかったゾ。こちらで修正してチェックしたけど，やはり内容が単調。アイデア面での工夫もほしい。

●Visual Basic用『GWAP』／RyuZyu（15歳）

編：説明文どおりに実行してみたのですが，マウスを動かしても文字が見えません。もう一度しっかり動作テストを行ってから再投稿してください。

●Visual Basic用『火の車』／よしとも（17歳）

Dr.D：マウス動作に連動させてキャラを動かすという処理はおもしろいが，キャラの動きがイマイチ安定していない感じじゃ。スタートから坂道だと，前になかなか進めないため，プレイしていてストレスがたまってしまう。今回，編くんが述べている“ゲーム・バランス”が悪いと言える。

●Visual Basic用『WinMenuControl』／小澤修一（18歳）

影：ベーマガでは珍しいActiveXコントロールとしての投稿。VB作品のメニュー表示機能を拡張することが可能なコントロール。レベルは高い作品なんだろうけど，扱うという面ではチョット難しいかな，という印象。実用性がある拡張とは言い難いため，無理して使用する人は少ないかも？

●C++Builder用『モザイクコンバータ』／山崎啓吾（15歳）

影：以前VBで掲載した同名作品のC言語版。Cへの移植は評価できるけど，VB版にあった機能が大幅に削減されているのが残念。このモザイク画像を作成するのには欠かせない，アンドゥ機能がないのは痛いな～。

●PC-E650用（実用）『FGS SYSTEM UTILITY』／Shin.Y（17歳）

つぐ美：ポケコンのファンクション・キーの拡張と言えるプログラム。実用プログラムで言うのも何ですが，チョットおもしろ味に欠ける内容なんですよね。個人的には，ポケコンならではのゲーム作品を見てみたいわね。

今月はもう一歩でした

●Visual Basic＋DirectX用『バドミントン』／橘　一馬（29歳）

編：DirectXの機能はしっかり生かしてはいるのですが，バドミントンゲームとして見ると，チョットいい加減な設定が多すぎます。

たとえば，サーブという行動がなく，シャトルが中央から上がってくるため，左クリックしながらシャトルを追えば100%COMに勝ててしまいますし，また，自分のコートでの2度打ちも可能になっています。

ラケットのレシーブ判定などがしっかりできていただけに，余計に残念でしたね。ルールやゲーム面でのしっかりした改良に期待したいですね。

Dr.Dの LABORATORY ベスト・プログラム研究室



今月のベスト・プログラム：VB用『Word Hunter』(42ページ掲載)

今月のベスト・プログラムは，VB用の『Word Hunter』。ゲーム内容自体はパズルという単純な形じゃが，パズルのピースをicoファイル化し，パソコンのディレクトリ内に保存するという処理が光っている作品じゃ。

今回は，このファイル関連の処理を重点的に解説していこう。

(Dr.D)

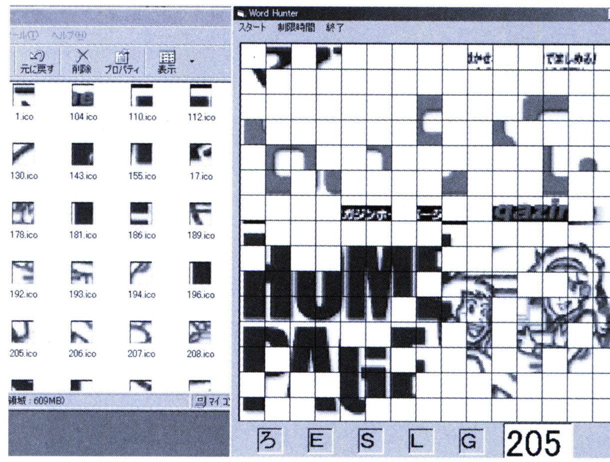
ファイル処理

ファイル処理と書いてはおるが，ファイルの保存ならびに読み込みにおいては，Load，Saveプロシージャ内でOpen命令を使用する単純な方法で実現されておる。この作品で肝となるファイル関連の処理はSerchプロシージャで行っているディレクトリ検索と言えるじゃろう。

このSerch内の処理，ファイル関連命令では，Dir，GetAttrといった命令があるが，重要になる処理は，じつはLeft$やMid$といった文字列関連の命令の部分なのじゃ。

なぜ，ディレクトリというファイル・システム関連の部分なのに文字列の処理が重要になるのかというと，ディレクトリを検索するのに，ディレクトリ名を文字列として変数に代入し，その変数の内容から検索を行っているためじゃ。では，その検索をどのように行っているかを解説していこう。

一つ目の検索となるのが，開始ディレクトリなどの初期設定を行ったあとに行うフォルダの検索じゃろう。



《写真2》 bmpファイル使用画面

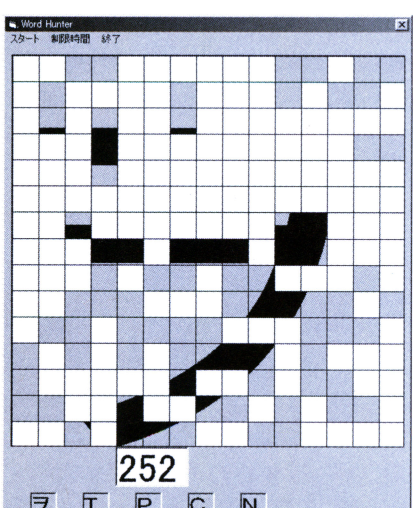
GetAttrで指定した名前がフォルダ(vbDirectory)であるかを確認し，かつそのフォルダが有効なもの(.や..フォルダでない)であれば，そのフォルダ内へ移動(現在のパスを示す文字列に¥を加えて指定したフォルダ名を連結させる)しているというものじゃ。単純なアルゴリズムじゃが，その分汎用性もある処理と言えるゾ。

つぎの検索となるのが，検索し終えたフォルダを示す文字列を配列で宣言している変数(FName)へコピーし，フォルダを一つ上へ戻すという処理じゃろう。

Fnameへのコピーは，ただ代入を行うだけじゃが，この代入処理こそSerch()プロシージャで行いたい処理。配列宣言のFnameへディレクトリのパスをコピーするため，配列の添字を乱数で指定するだけで，ランダムなフォルダへファイルを保存することができるわけじゃ。

フォルダを一つ戻す処理は，InStrRev命令を使用し，文字列を後ろから検索することで，文字列変数における現在のフォルダ名までの位置を判断し，そのフォルダ名を削除することで実現しておるのじゃ。

こうしてみると，命令面での高度な処



《写真1》 15×15への改造画面

《第1表》 マス目増加改造の手順

15×15マスにする場合
フォームプログラムの変更
Picture1_MouseDown()内の10の値を15に変更
モジュールプログラムの変更
For文で使用されている100の値を225に変更 For文やLineで使用されている320の値を480に変更 Syoki2()内のFontSizeの指定を400ぐらいに調整する Hozon()内の288の値を448に変更 DragDrop()内の10を15へ，320を480へ，それぞれ値を変更
フォームコントロールの設定
Picture1のHeight，Widthをそれぞれ7200へ変更。 それに合わせて，他のコントロールを表示されるように移動させる

理はしておらんが，文字列を扱うというアイデア面を駆使して，ディレクトリ検索を実現しておるのがわかるじゃろう。プログラムは，作品内容だけでなく，リストレベルにおいてもアイデアが重要だということじゃな。

BEST PROGRAM 改造してみよう

今回の改造は，リスト内の乱数部分の修正，パズルのマス数の増加，bmpファイルをパズル化する，という3点を行ってみたゾ。

まず，乱数部分の修正というのは，Syoki3プロシージャで行っている“回答欄が重複していたときの処理”という部分のことじゃ。この部分の記述があまりスッキリしていないため，チョット修正したリストを作成してみた。参考にしてほしい（**リスト1**）。文字列でなく，乱数の値で比較を行っているのがポイントじゃな。

つぎは，説明文にも書かれていたマス目の増加の改造じゃ。実現したリストを載せたいところじゃが，誌面の関係でそれはムリ。そこで**第1表**に実現するための方法をまとめてみた。Picture1の大きさに関連する数値の変更や，大きさに合わせたファイル分割の数値を変更していけば，実現できるじゃろう（**写真1**）。

三つ目の改造は，表示されるパズルを文字ではなくbmpファイルの画像にしようというもの（**写真2**）。実際これは表示する画像ファイルとそれと同じ大きさの真っ白な画像ファイル（cls.bmp）さえ用意すれば**リスト2**のような簡単なコーディングで実現可能じゃ。

しかし，このままでは見栄えがする形にはなるが，このゲームで用意されている文字列と，当てるという処理が使えないため，ゲームとしては成立しない。

そのため，本格的に改造したいのであれば，ゲーム・システムそのものの改造が必要になる。これについては，興味を持った読者自身の手で行ってみてほしい。パズルを完成させればクリア，というふうにするもよし，今のシステムのままで行うのであれば，複数の小さな画像ファイルを用意して表示するなど，いろいろな改造があるはずじゃ。

《リスト1》乱数部分の修正リスト

```
I = Int(Rnd * 4) + 1
 tempA = I
 A(I) = A(0)
 For I = 1 To 5
 If I <> tempA Then
   tempX(I) = Int(Rnd * 3)
   tempR(I) = Int(Rnd * Len(M(tempX(I)))) + 1
   For J = 1 To I - 1
    If tempX(J) = tempX(I) Then
        If tempR(J) = tempR(I) Then
            tempR(I) = ((tempR(I) + 1) Mod Len(M(tempX(I)))) + 1
            J = 0
        End If
    End If
   Next
   A(I) = Mid$(M(tempX(I)), tempR(I), 1)
   Form1.Picture3(I).Print A(I)
 End If
 Next
 With Form1
   .Picture3(tempA).Cls
   .Picture3(tempA).Print A(tempA)
 End With
```

《リスト2》画像ファイル表示リスト

```
Sub Syoki2()
 (途中省略)
With Form1
   .Picture1.Picture = LoadPicture("bmlogo.bmp")
   .Picture1.Visible = False
 End With
 (以下省略)

Sub Hozon()
 (途中省略)
With Form1.Picture1
 .Picture = LoadPicture("cls.bmp")
 .Visible = True
 .BackColor = vbWhite
End With
 (以下省略)
```



PlayStation 2でゲームが作れる！

「BASIC STUDIO パワフルゲーム工房」入門

第5回 RPGを作ってみよう

文／山崎　寛

今回はBASIC STUDIOでRPGを作ってみましょう。RPGといっても本格的なものではなく，簡単な内容です。気楽に挑戦してみてください。先月までの内容（マップ上でのキャラクター移動）もベースにしてますから，わからなくなった方は読み返してみましょう。

『BASIC STUDIO パワフルゲーム工房』
発　売：アートディンク
対応機種：PlayStation 2
メディア：CD-ROM 2枚組
発売中
価　格：9,800円（税抜）
http://www.artdink.co.jp/japanese/title/bs/index.html



どんなゲームを作るのか

今回はちょっとゲームらしく，RPGを作ってみることにします。

実際にプログラムを作りはじめる前に，簡単に作るゲームの内容を決めてしまいましょう。RPGといっても色々なものがありますが，今回作ろうとするゲームは

- **マップを移動していると，ときどき敵（モンスター）が出現**
- **戦闘はコマンド（戦闘，逃げる）を選び，自分，敵が順番に攻撃する**
- **自分，モンスターそれぞれに体力，攻撃力，防御力といったパラメータを持たせ，攻撃の際はそれらの数値から導いた値を体力から引く**
- **自分にはレベル，経験値のパラメータを持たせ，敵を倒して経験を貯めるとレベルアップ，体力などを上昇させる**
- **特定の場所（スタート地点）で体力が回復する**

と，実際にゲームにするにはまだ決めることがありますが，大体このような感じで作ってみることにしましょう。

プログラムの流れを考えよう

内容が決まったらプログラムに入りますが，その前にプログラムがどんな風な流れで動作するのかを考えてみましょう。**第1図**のように，ゲームの流れと照らし合わせながら流れを書きだしてみれば，どんな内容をプログラムしたらいいか整理できます。

プログラミングに慣れていれば，何も用意せず小規模のプログラムをいきなり組み始めても問題ありませんが，初心者のうちはいきなりプログラムをしようとしても「ゲームを作りたいけれど，どうプログラムしていいかわからない」状態に陥ってしまいがちです。ですが，プログラムの流れを書きながら整理することで，こうしたことを防ぎやすくなります。

これが今回のサンプルRPGだ

リスト1が今回サンプルとして作成したRPGです。ずいぶん簡略化しましたが，一応キャラクターが戦闘を繰り返して成長していくという内容になっています。

実行すると**写真1**のようにマップ上を移動できます。スタート地点には家があり，ここを通ると体力が全部回復します。画面中央少し右にあるのがボスの居所で，これを倒すのが目的となります。

移動画面では，パッドで上下左右に主人公を動かし，○ボタンを押すと画面右下に現在のパラメーターが表示されます（**写真2**）。ボタンを離すと表示は消えます。

マップ上をうろうろしていると，ときどき敵が現れます（**写真3**）。敵のグラフィックは表示されません。表示が欲しい方は，テクスチャに敵の姿を用意して，スプライトで表示するように改造するといいでしょう。

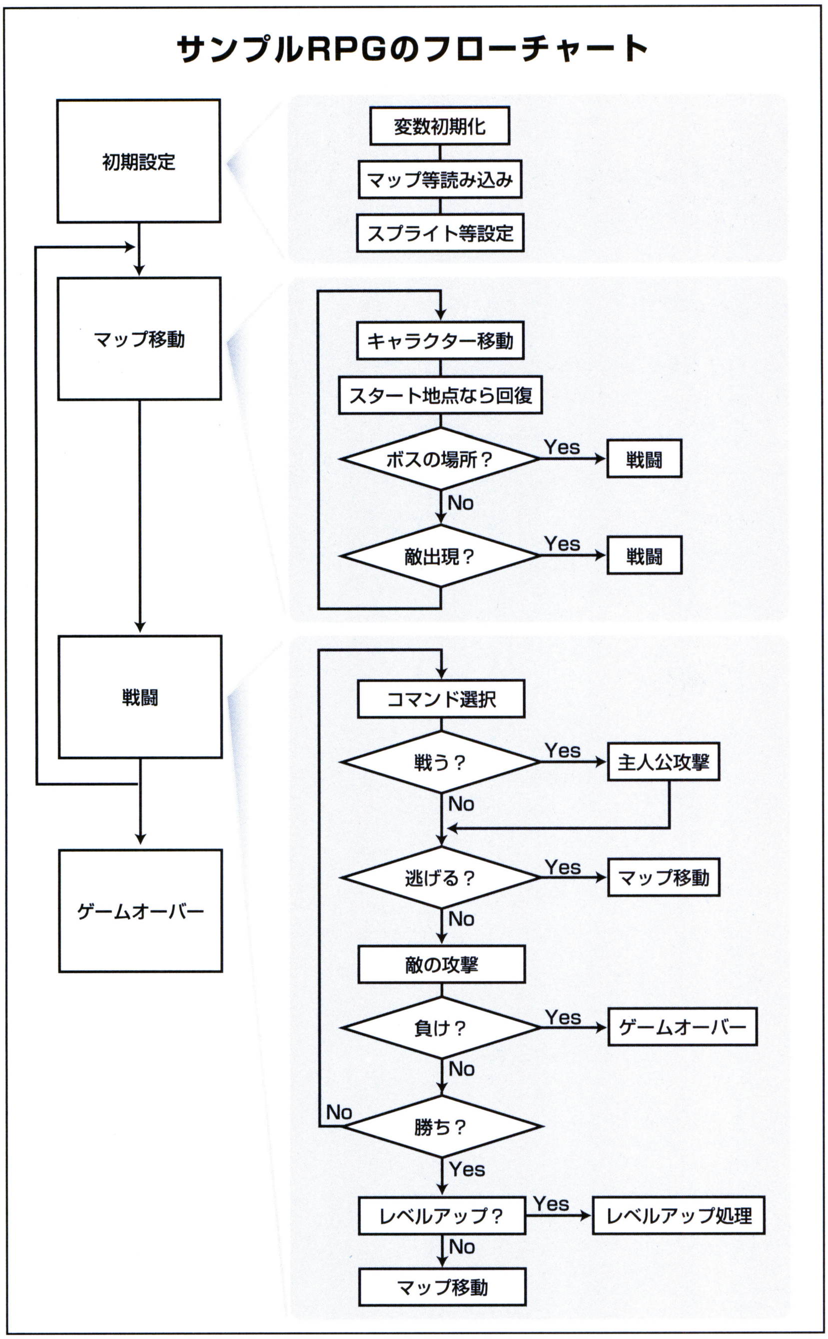
敵との戦闘は○ボタン，×ボタンで戦うか逃げるかを選択して，主人公，敵の順番でお互いにダメージを与え合っていきます。敵を倒せれば経験値が入り，繰り返すとレベルが上がりだんだんと主人公が強くなっていきます。

と，内容的には一般的なコンピュータRPGから，レベル上げの面倒な作業のみを抽出したようなものとなっています。これが面白いかどうかはともかく（笑），RPGを作りたい人の参考にはなるかと思います。

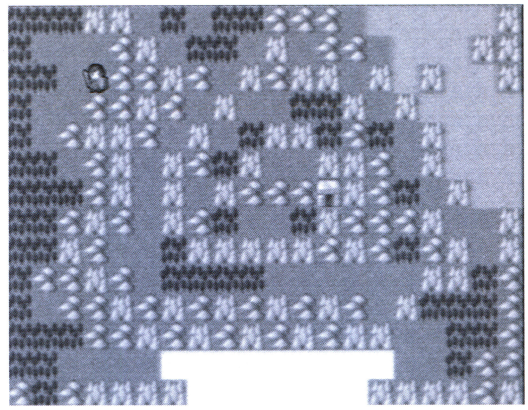
プログラムの解説

リスト1ですが，リストを見ただけではわかりにくい部分もあるかと思うので，そうした部分を簡単に解説したいと思います。

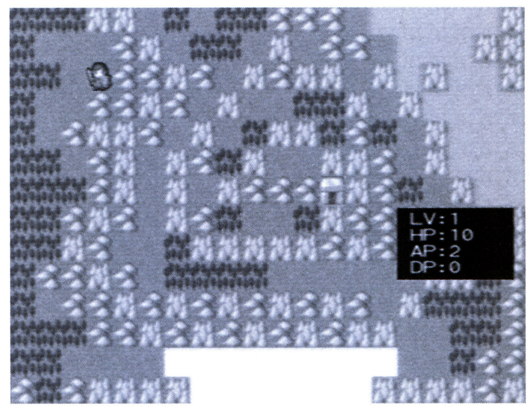
このプログラムでは，マップの地形の並びや場所ごとにどんな敵が出現するか，敵の名前や各パラメータなどがプログラム末尾のDATA文にまとめられています。DATA文の後ろの内容は，READ文を実行するごとに順番に読み，自



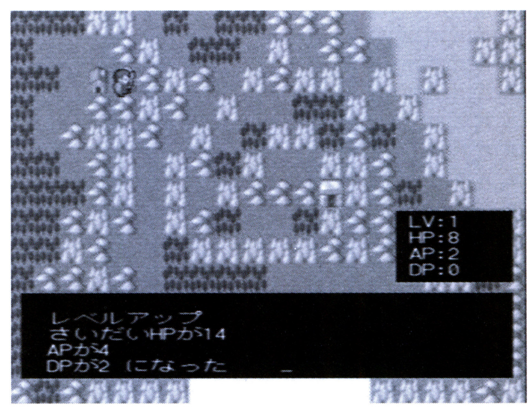
《第1図》全体の流れをおおざっぱに考えてから，だんだんと細かい部分を整理する



《写真1》主人公は左上の方にいる。そこからボスのいる中央の入口まで移動していく



《写真2》○ボタンを押すとパラメータが表示されるのだ



《写真3》敵との戦闘。下に戦闘の経過が表示される。ちゃんとレベルアップして，主人公が強くなっていくのだ

動的に変数へと入ります。このプログラムでは初期設定の部分にあるREAD文によって，それぞれのデータを配列変数へ入れています。このように，DATAとREADを使うことでデータを内容をわかりやすく効率的に扱うことができます。

マップのデータは，20文字を14個並べてありますが，これは1文字が画面の地形一つと対応していて，この数字がマップの種類を示しています。読み込む際には横1列をREADで読んでから，1文字ずつ分解して配列に入れています。この処理は460行で行っており，この中にはちょっと見慣れない

```
A$(I+1:I+1)
```

などという表記がされています。

これは文字列の一部を取りだす表記で，従来のBASICを知っている方にはMID$の代わりだと言えばわかりやすいでしょう。

この表記では，A$の中から取りだす文字を何文字目から何文字目までという形で指定します。たとえば，A$(2:3)なら2文字目から3文字目までということです。

マップデータの後に同じようなデータがありますが，これはマップのどの位置にどの種類の敵がでてくるかを示したもので，マップと1対1に対応しています。この番号が大きいほど強い敵がでてくるようにしてあり，これによってスタート地点に近いほど弱い敵，遠いほど強い敵が出現します。

主人公のパラメータ表示や，戦闘の際のメッセージ表示などは，PRINT文で

テキストを表示しているのですが，そのままではマップと重なって読みにくいため，タイルプリミティブで黒い四角を表示しています。タイルプリミティブというのは，スプライトのテクスチャがないようなもので，単色の四角形を自由に重ね合わせて表示できます。このプログラムでは，ステータス用に番号1を，メッセージ用に番号2のタイルプリミティブを使っています。

1350行～1450行，1470行～1600行のように，「SUB ○○～END SUB」という形の部分をサブルーチンといいます。この部分はプログラムの好きな場所からCALL ○○とすることで実行され，GOTO文と違ってサブルーチンの内容を終えると自動的に戻るようになっています。このため，プログラムの違う部分から何度も使うような処理をサブルーチンにしておくとプログラムをすっきりと整理できます。

まとめ

このゲームでは「ボタンが押されるまで待つ」という場合がありますが，このときボタンが押しっぱなしになっていた場合には，一度ボタンを離して再度押されない限り反応しないようにしています。これは，BTNという変数を用意して，ボタンが一度押されたらつぎに放されるまでBTNに1を入れ，ボタンの状態をチェックする部分ではボタンの状態と共にBTNを調べることで実現しています。細かい部分ですが，こうした処理をするかどうかでゲームの操作性，遊びやすさがずいぶん変わります。

あと，主人公や敵の各パラメータや，各レベルに必要な経験値，レベルアップに伴う上昇値などはかなり適当な値や単純な計算となっています。一応最初は弱い敵と対等で，だんだんと強い敵と戦えるようなバランスにしてありますが，自分なりにバランスを変えてみるのも面白いかもしれません。

《リスト1》サンプルRPGプログラム・リスト

```
100 !●●初期設定●●
110 INIT
120 DIM P(4),MAP(0 TO 19,0 TO 13)  ――マップ
130 DIM MPX(0 TO 7),MPY(0 TO 7)  ――テクスチャ用
140 DIM MMAP(0 TO 19,0 TO 13)  ――モンスター分布図
150 DIM MN$(20),MMHP(20),MA(20),MD(20),ME(20)  ――各モンスターのパラメータ
160 P(1)=0
170 P(2)=32
180 P(3)=0
190 P(4)=64    } 主人公のポーズ用
200 X=96
210 Y=32
220 PO=1
230 MU=0
240 BTN=0  ――ボタン押しっぱなし対策用
250 !●主人公のパラメータ設定
260 MHP=10  ――体力
270 LV=1
280 AP=2
290 DP=0
300 EX=0
310 !●スプライト関連初期化
320 SPRPAL 1,11,0,1
330 SPRDISP 1
340 TILECOL 1,0,0,0,128
350 TILECOL 2,0,0,0,128
360 TILEPOS 1,480,114,624,154,2
370 TILEPOS 2,16,160,624,208,2    } ステータス，メッセージ用タイルプリミティブで黒い四角を表示
380 !●マップ用テクスチャ位置情報を読み込む
390 FOR I=0 TO 7
400   READ MPX(I),MPY(I)  ←――データⒶを読みこむ
410 NEXT I
420 !●マップを読みこみ表示する
430 FOR J=0 TO 13
440   READ A$  ←――データⒷを読みこむ
450   FOR I=0 TO 19
460     R=VAL(A$(I+1:I+1))  ――データを1文字ずつ分解
470     MAP(I,J)=R  ――マップ用配列にマップの種類を入れる
480     K=J*20+I+100  ――スプライト番号を求める
490     SPRPAL K,11,0,1
500     SPRDISP K
510     SPRPOS K,I*32,J*16,0
520     SPRTEX K,9,MPX(R),MPY(R),MPX(R)+32,MPY(R)+32    } スプライトの設定
530   NEXT I
540 NEXT J
550 !●敵出現マップを読み込む
560 FOR J=0 TO 13
570   READ A$
580   FOR I=0 TO 19
590     MMAP(I,J)=VAL(A$(I+1:I+1))  ←――データⒸを読みこむ
600   NEXT I
610 NEXT J
620 !●敵パラメータを読み込む
630 FOR I=1 TO 20
640   READ MN$(I),MMHP(I),MA(I),MD(I),ME(I)  ←――データⒹを読みこむ
650 NEXT I
660 !●●マップ移動●●
```

```
670 LABEL IDOU
680 DO
690   TX=X
700   TY=Y
710   IF PAD(1,4)>0 THEN
720     MU=0
730     PO=MOD(PO,4)+1
740     Y=Y+2
750   END IF
760   IF PAD(1,2)>0 THEN
770     MU=96
780     PO=MOD(PO,4)+1
790     X=X-4
800   END IF    } パッド操作に応じて主人公を動かす
810   IF PAD(1,3)>0 THEN
820     MU=192
830     PO=MOD(PO,4)+1
840     Y=Y-2
850   END IF
860   IF PAD(1,1)>0 THEN
870     MU=288
880     PO=MOD(PO,4)+1
890     X=X+4
900   END IF
910 !●移動できない場所では元の位置に戻す
920   FOR I=0 TO 1
930     FOR J=0 TO 1
940       IF X<0 OR X>608 OR Y<0 OR Y>208 OR MAP(INT((X+I*31)/32),INT((Y+J*15)/16))=3 THEN
950         X=TX
960         Y=TY    } 画面端，通れない場所へ移動した場合，元の位置に戻す
970       END IF
980     NEXT J
990   NEXT I
1000 !●スタート地点で体力回復
1010   IF INT((X+16)/32)=3 AND INT((Y+8)/16)=2 THEN HP=MHP  ――最大HPへ戻す
1020 !●ボタンでパラメータ表示
1030   IF PAD(1,6)>0 THEN
1040     CALL DISP
1050     BTN=1    } ボタンが押されている場合，ステータス表示
1060   ELSE
1070     BTN=0
1080     TILEOFF 1
1090     CLEAR    } 押されていない場合，表示をかくす
1100   END IF
1110 !●ボス出現
1120   IF INT((X+16)/32)=12 AND INT((Y+8)/16)=6 THEN
1130     M$="ボス"
1140     MH=999
1150     MAT=100    } ボスのパラメータを入れて戦闘へ
1160     MDF=50
1170     GOTO *BATTLE
1180   END IF
1190 !●敵出現
1200   IF X<>TX OR Y<>TY AND INT(RND*40)=0 THEN  ――主人公が移動し，乱数の値が規定値なら敵出現
1210     MS=INT(MMAP(INT((X+16)/32),INT((Y+8)/16))*2+RND*2)+1  ――敵分布データと乱数を元に種類を決める
1220     M$=MN$(MS)
1230     MH=MMHP(MS)
```

```
1240   MAT=MA(MS)
1250   MDF=MD(MS)        パラメータを入れて戦闘へ
1260   MEX=ME(MS)
1270   GOTO *BATTLE
1280  END IF
1290 !●主人公表示
1300  SPRPOS 1,X,Y,1
1310  SPRTEX 1,9,MU+P(PO),0,MU+P(PO)+32,32
1320  FSYNC 3
1330 LOOP
1340 !●●パラメータ表示●●
1350 SUB DISP
1360 TILEDISP 1 ――― 黒いバックを表示
1370 LOCATE 30,12
1380 PRINT "LV:";LV
1390 LOCATE 30,13
1400 PRINT "HP";HP
1410 LOCATE 30,14        各パラメータを文字表示
1420 PRINT "AP:":AP
1430 LOCATE 30,15
1440 PRINT "DP:";DP
1450 END SUB
1460 !●●ボタン入力待ち●●
1470 SUB WAIT
1480 DO
1490  LOCATE 20,21
1500  IF T<30 THEN PRINT "_"; ELSE PRINT " ";
1510  T=T+1        メッセージの下に記号を点滅表示
1520  IF T>60 THEN T=0
1530  FSYNC 1
1540  IF PAD(1,6)=0 THEN BTN=0
1550 LOOP WHILE PAD(1,6)=0 OR BTN=1 ――― ボタンが押されるまでループ
1560 BTN=1
1570 CLEAR
1580 CALL DISP
1590 LOCATE 0,18
1600 END SUB
1610 !●●戦闘●●
1620 LABEL BATTLE
1630 !●コマンド選択
1640 TILEDISP 2
1650 LOCATE 0,18
1660 PRINT " ";M$;"があらわれた"
1670 LABEL BATTLE2
1680 PRINT " どうする？"
1690 PRINT " ○：たたかう"
1700 PRINT " ×：にげる"
1710 DO
1720  IF PAD(1,6)=0 THEN BTN=0
1730 LOOP WHILE PAD(1,6)+PAD(1,7)=0 OR BTN=1 ――― ○か×が選ばれるまでループ
1740 BTN=1
1750 CLEAR
1760 CALL DISP
1770 LOCATE 0,18
1780 IF PAD(1,6)>0 THEN
1790  PRINT " プレイヤーのこうげき"
1800  A=INT(AP*(RND*0.8+1)-MDF)
1810  IF A<0 THEN A=0        ○が押されたら主人公の攻撃
1820  PRINT " ";M$;"に";A$;"のダメージ"
1830  MH=MH-A
1840  IF MH<=0 THEN GOTO *LEVEL ――― 倒した
1850 END IF
1860 IF PAD(1,7)>0 THEN
1870  PRINT " ";M$;"からにげだした";
1880  FOR I=0 TO 3
1890   PRINT ".";
1900   FSYNC 60        ×が押されたら逃げる
1910  NEXT I
1920  IF RND<0.7 THEN ――― 乱数で逃げられるか判定
1930   TILEOFF 2
1940   GOTO *IDOU
1950  END IF
1960  PRINT
1970  PRINT " がにげられなかった"
1980 END IF
1990 !●敵の攻撃
2000 CALL WAIT
2010 PRINT " ";M$;"のこうげき"
2020 A=INT(MAT*(RND*0.8+1)-DP)
2030 IF A<0 THEN A=0
2040 PRINT " ";A;"のダメージ"
2050 HP=HP-A
2060 IF HP<=0 THEN GOTO *OVER ――― やられた
2070 CALL WAIT
2080 GOTO *BATTLE2
2090 !●敵を倒した
2100 LABEL LEVEL
2110 PRINT " ";M$;"をたおした"
2120 CALL WAIT
2130 IF M$="ボス" THEN
2140  PRINT " ゲームクリアおめでとう"        ボスを倒した
2150  END
2160 END IF
2170 PRINT " けいけんち";MEX;"をかくとく"
2180 CALL WAIT
2190 EX=EX+MEX
2200 IF EX>(LV+1)^3 THEN
2210  PRINT " レベルアップ"
2220  LV=LV+1
2230  A=LV^2+10
2240  PRINT " さいだいHPが";A
2250  MHP=A
2260  A=LV*2        レベルアップ
2270  PRINT " APが";A
2280  AP=A
2290  A=INT(LV*1.2)
2300  PRINT " DPが";A;"になった"
2310  DP=A
2320  CALL WAIT
2330 END IF
2340 TILEOFF 2
2350 GOTO *IDOU
2360 !●ゲームオーバー
2370 LABEL OVER
2380 CALL *WAIT
2390 PRINT " あなたはやられてしまった"
2400 PRINT " ゲームオーバー"
2410 END
3000 !●●マップ用テクスチャ位置データ●●
3010 DATA 384,0,416,0,448,0,480,0,384,32,416,32,448,96,480,96 ――― データⒶ
3020 !●●マップデータ●●
3030 DATA 11133010112220555553
3040 DATA 10002301103022055533
3050 DATA 11072232032330353553
3060 DATA 10022320300113035555
3070 DATA 10233203013312103555
3080 DATA 11023032130033203555
3090 DATA 11123030322263210355
3100 DATA 11232032100132223003        データⒷ
3110 DATA 11320013333323212303
3120 DATA 12232011110000003312
3130 DATA 10223232323232031112
3140 DATA 11122323232323300112
3150 DATA 11000044444444400122
3160 DATA 21233334444444333323
3170 !●●敵分布図データ●●
3180 DATA 11110777777777777666
3190 DATA 10000077777777777666
3200 DATA 10000008879777776666
3210 DATA 10000088899997766666
3220 DATA 11110888999999996666
3230 DATA 11111889999999996666
3240 DATA 21122889999998888666
3250 DATA 22228889999988886666        データⒸ
3260 DATA 22288888888888888666
3270 DATA 22228888888888886666
3280 DATA 32233848484858856556
3290 DATA 33333344444455555555
3300 DATA 33333444444455555555
3310 DATA 33333444444555555555
3320 !●敵の名前とパラメータ
3330 DATA スライム,3,1,0,1
3340 DATA コボルト,5,2,0,2
3350 DATA ゾンビ,7,2,1,4
3360 DATA バット,8,3,2,7
3370 DATA ゴブリン,10,4,2,11
3380 DATA オーク,13,6,3,16
3390 DATA グール,16,8,3,22
3400 DATA トロル,20,10,4,29
3410 DATA ゴースト,30,12,5,37
3420 DATA リッチ,40,15,6,46
3430 DATA ゴーレム,55,18,7,56        データⒹ
3440 DATA マミー,70,21,8,67
3450 DATA キメラ,88,25,9,79
3460 DATA ハーピー,105,29,10,92
3470 DATA スケルトン,128,34,11,106
3480 DATA メデューサ,150,39,12,121
3490 DATA サイクロプス,170,45,13,137
3500 DATA ミノタウロス,190,52,14,154
3510 DATA ヴァンパイア,220,59,15,172
3520 DATA ドラゴン,240,70,16,191
```



ワンダースワンのソフトを自作しよう!

チャレンジ!! ワンダーウィッチ Challenge WonderWitch

ワンダースワン ホビープログラミングツール WonderWitch ワンダーウィッチ

第11回「割り込み処理に挑戦」



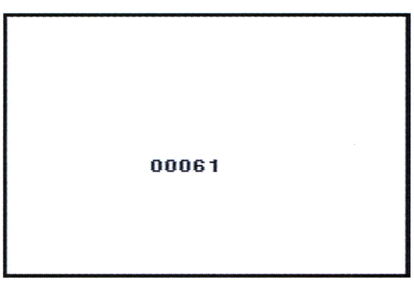
ワンダースワン・ホビープログラミングツール
「ワンダーウィッチ(WonderWitch)」
定価16,800円(税別)
◆パッケージ内容
・PC-WonderSwan 接続ケーブル
(スワンケーブル,クロスケーブル)
・ワンダーウィッチ専用カートリッジ
・CD-ROM ・マニュアル
◆動作環境
機種: PC/AT 互換機
OS: Windows 98/Me/NT4.0/2000
CPU: Pentium 133MHz以上
メモリ: 32MB以上(推奨64MB以上)
ディスプレイ: 640×480, 256色以上
シリアルポート: 9600/38400bps
※専用カートリッジのみ単体の購入も可能。
3,980円(税別)。
◆製品について問い合わせ
wonderwitch-info@bandai.co.jp
◆通信販売について問い合わせ
netshop@bandai.co.jp

連載1年を経て,テーマというテーマを出し尽くした感のある「チャレンジ!!ワンダーウィッチ」ですが,「割り込み」についてのレクチャーを忘れていました。というわけで,今回は割り込みを使ったゲームの紹介です。

(松原拓也)

割り込みとは?

「割り込み」というのは,今行っている処理を一旦止めて,その途中から別の処理を割り込ませる技術です。ワンダーウィッチではシステム制御BIOSで8種類の割り込みをサポートしています(第1表)。



▲《画面1》割り込みを使ったストップウォッチの実行画面。1秒間隔で値が増え続ける

これらの割り込みはスリープ中でも有効にすることができます。

そこで,「WonderWitch デベロッパーズ マニュアル」の「Freyaシステム制御BIOS仕様」を参考にして,簡単なサンプルを作ってみました。「割り込みを使ったストップウォッチ(swatch.fx)」です(リスト1,画面1)。ワンダースワンでは「VBLANK期間開始」の割り込みが1秒間に75回ほど発生しますので,この割り込みの回数をもとに1秒間をカウントしています。

割り込み用BIOSの使用方法

では,サンプル(リスト1)を例に具体的な使いかたを説明します。割り込みは「割り込みフックセット関数(sys_interrupt_set_hook)」で設定します。書式は以下のとおりです。

```
void near sys_interrupt_set_hook(int type, intvector_t far *vectorNew, intvector_t far *vectorOld);
```

・引数typeは「SYS_INT_VBLANK(VBLANK期間開始)」に設定します。

・vectorNewは今回設定する割り込みハンドラです。vectorOldは退避用の前回設定されていたハンドラです。引数をして使う割り込みハンドラ構造体は以下のような形式になっています。

```
struct intvector_t {
  void near (*callback)();
  word cs;
  word ds;
  word reserve;
}
```

・callbackは割り込みで実行させたい関数のアドレスです。ポインタはnearになっていますが,実際に呼ばれる側の関数はfarポインタである必要があります。

・cs(コードセグメント)とds(データセグメント)は「_asm_inline」関数を使い,アセンブラ表記で直接CSとDSを取り出します。

・reserveは無視してかまいません。

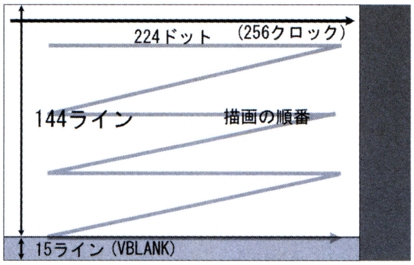
なお,プログラムの最後では「割り込みフックリセット(sys_interrupt_reset_hook)」を使い,退避させた割り込みハンドラを元に戻してあげましょう。

《第1表》割り込みの一覧表

値	シンボル	割り込み内容
0	SYS_INT_SENDREADY	シリアル送信データ・エンプティ
1	SYS_INT_KEY	キー割り込み
2	SYS_INT_CASETTE	カセット内のRTCアラーム割り込み
3	SYS_INT_RECEIVEREADY	シリアル受信データ・レディー
4	SYS_INT_DISPLINE	描画ライン番号検知割り込み
5	SYS_INT_VBLANK_COUNTUP(SYS_INT_TIMER_COUNTUP)	VBLANKタイマー・カウント終了
6	SYS_INT_VBLANK	VBLANK期間開始
7	SYS_INT_HBLANK_COUNTUP	HBLANKタイマー・カウント終了

HBLANK・VBLANKとは

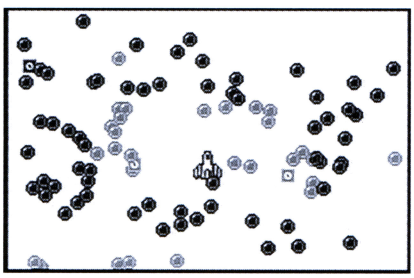
第1図を見てください。ワンダースワンは画面を,上から下に1ラインずつ描いて処理しています。1画面は144ラインですが,そのあとに見えない余白を15ライン分持っていて,この余白を「VBLANK」



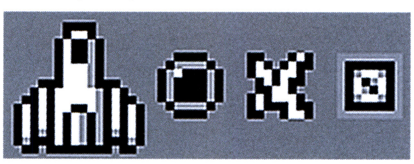
▲《第1図》VBLANKとHBLANKのイメージ図

といいます。VBLANK1回分の周期は13.25mSecで，1秒間に約75.47回「VBLANK期間開始」割り込みが発生します。

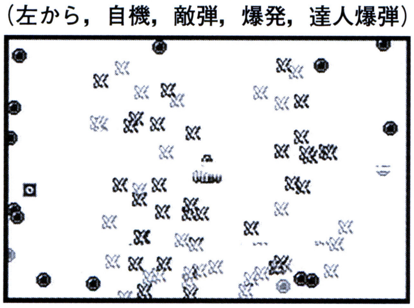
また，1ライン分の描画時間はCPUクロック256回分で，これが「HBLANK」の周期です。時間にして83.333…uSec。



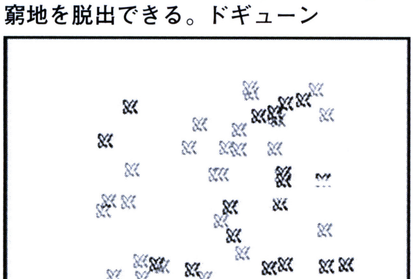
▲《画面2》飛び交う弾の移動をVBLANK割り込みで処理。あと，一度に200個以上のスプライトを出す実験もしてみました



▲《第2図》ゲーム中のキャラクター（左から，自機，敵弾，爆発，達人爆弾）



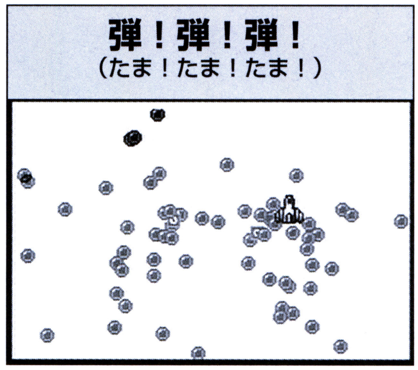
▲《画面3》Bボタンの「達人爆弾」で窮地を脱出できる。ドギューン



▲《画面4》一定回数，自機に弾が当たると大爆発してゲーム・オーバー

1秒間に約12,000回「HBLANKタイマー・カウント終了」割り込みが発生します。

割り込みを使ったゲーム



自機を操って，雨あられと降り注ぐ敵弾をかいくぐるゲームです（**リスト2，画面2，第2図**）。もし自機がピンチになったら「達人爆弾」で周りを一掃できます（**画面3**）。「達人爆弾」は敵弾にまぎれて飛んでいますので，接触して回収することができます（スタート時は3発）。一定のダメージを受けたら大爆発してゲーム・オーバーです（**画面4**）。

弾の移動処理を「VBLANK期間開始」割り込み（1/75秒周期）で行うように作ってみました。これによってメイン・プログラムの動作速度に依存せずに弾の移動速度を固定することができます。

また，今回のプログラムではスプライトを高速に切り換えて，一度に200個以上のスプライトを表示できないか実験してみました。しかし，結果は見た目がチカチカしてしまい，あまり成功ではありません。

スプライトは通常どおり128個以内にとどめておいたほうがよさそうです。

まとめ

無理してVBLANK割り込みを使わずに「sys_wait」でも同期を取るようなプログラムを作ることができます。そこで，もっと割り込みにしかできないような使い道（たとえば，スリープ状態からカセット割り込みで復帰するような荒業）を研究してみるとおもしろいかもしれません。

（※8月号で告知した「ワンダーゲート」プレゼントは10月号で発表します。お楽しみに）

《リスト1》“swatch.c”（WonderWitch）

```
/*--------------------割り込みを使ったストップウォッチ by 松原 */

#include    <sys/bios.h>


/*---------------------割り込みプログラム*/
void far intr_vblank()
{
    static int iCnt = 0;
    static int iSec = 0;

    iCnt = (iCnt + 1) % 75;
    if(iCnt == 0)
        text_put_numeric(10,10 ,5 ,NUM_PADZERO ,iSec++);
}

/*---------------------メインプログラム*/
void main ()
{
    intvector_t vectorNew;
    intvector_t vectorOld;

    text_screen_init();

    vectorNew.callback = (void (near *)())FP_OFF(&intr_vblank);
    vectorNew.cs = _asm_inline("    mov    ax,cs");
    vectorNew.ds = _asm_inline("    mov    ax,ds");
    sys_interrupt_set_hook(SYS_INT_VBLANK, &vectorNew, &vectorOld);

    while (1) {
        if( key_hit_check_with_repeat() & KEY_START)
            break;
    }

    sys_interrupt_reset_hook(SYS_INT_VBLANK, &vectorOld);
}
```

《リスト2》 “tama.c”（WonderWitch）

```
/*-----------------弾！弾！弾！    by 松原 */

#include <sys/bios.h>
#include <stdlib.h>


char cChrDat[]={
 /*自機*/

0x00,0x03,0x01,0x03,0x02,0x07,0x02,0x07,0x02,0x07,0x03,0x07,0x03,0x07,0x
03,0x1F,

0x00,0xC0,0x80,0xC0,0x40,0xA0,0x40,0xA0,0x40,0xA0,0xC0,0xA0,0xC0,0xA0,0x
C0,0xB8,

0x17,0x3F,0x17,0x7F,0x56,0xFF,0x55,0xFF,0x55,0xFF,0x55,0xEF,0x45,0xBF,0x
00,0x6F,

0xE8,0xDC,0xE8,0xDE,0x6A,0xDF,0xAA,0x5F,0xAA,0x5F,0xAA,0x57,0xA2,0x5D,0x
00,0xF6,
 /*弾*/

0x00,0x3C,0x3C,0x42,0x62,0xBD,0x42,0xBD,0x42,0xBD,0x42,0xBD,0x3C,0x42,0x
00,0x3C,
 /*爆発*/

0x02,0xC7,0x44,0xEF,0x6C,0xFE,0x28,0x7E,0x18,0x7C,0x5C,0xFE,0x44,0xFF,0x
06,0xCF,
 /*パワー補給*/

0xFF,0x00,0x81,0x7E,0xBD,0x5A,0xAD,0x76,0xB5,0x6E,0xBD,0x5A,0x81,0x7E,0x
FF,0x00
};

#define SPN_MIN   0      /*スプライト番号：*/
#define SPN_TAMA 0       /*スプライト番号：敵弾*/
#define SPN_TAMAMAX 124 /*スプライト番号：一度に出せる敵弾*/
#define SPN_MY   124    /*スプライト番号：自機*/
#define SPN_MAX 128     /*使用スプライト数*/

#define CHN_MIN   0  /*キャラクター番号：*/
#define CHN_MY 0  /*キャラクター番号：自機*/
#define CHN_TAMA 4   /*キャラクター番号：敵弾*/
#define CHN_BOMB 5   /*キャラクター番号：爆発*/
#define CHN_POW   6   /*キャラクター番号：爆弾*/

#define CPN_SP 8  /*キャラクターパレット番号：スプライト有効開始値(常数)*/
#define CPN_MAS 12   /*キャラクターパレット番号：透明色有効(常数)*/
#define  CFM_CPN  ((CPN_MAS-CPN_SP)<<  CFSFT_PALETTE)    /*パレット番号割り振り
*/

#define CHR_PIX8  /*  1キャラクターの1辺の長さ[pixel]*/
#define CHR_BYTE 16 /*  1キャラクターのバイト数(4階調時) */
#define WIDTH_TAMA (CHR_PIX)
#define WIDTH_MY (CHR_PIX * 2)

#define TAMA_NONE  999

#define TAMAMAX 220   /*表示される最大敵弾数*/

typedef struct {
   int x;
   int y;
   int x1;
   int y1;
   int chn;
} ST_TAMA;   /*敵弾情報  構造体*/


ST_TAMA tama[TAMAMAX]; /*敵弾情報*/
int iMx=(LCD_PIXEL_WIDTH/2);   /*自機の座標*/
int iMy=(LCD_PIXEL_HEIGHT/2);
int iBomb=3;   /*自機の爆弾数*/
int iMhp=20;   /*自機のHP*/

/*-------絶対値を求める関数----*/
/*引数（比較したい値1 ,比較したい値2）*/
int fnc_abs(int d1,int d2)
{
    if(d1 > d2)
        return(d1-d2);
    else
        return(d2-d1);
}

/*-------------符号を求める関数----*/
/*引数（符号を求めたい値）*/
int fncSgn(int d1)
{
    if(d1 > 0)
        return(1);
    else
        return(-1);
}

/*------------------------敵弾の初期化*/
void subTamaInit()
{
    int iId;

    for(iId=0 ;iId<TAMAMAX ;iId++) {
        tama[iId].x = TAMA_NONE;
        tama[iId].chn = CHN_TAMA;
    }
}

/*------------------------敵弾の発射*/
void subTamaFire(int iTx ,int iTy,int iTx1 ,int iTy1 )
{
    int iId;

    for(iId=0 ;iId<TAMAMAX ;iId++) {
        if(tama[iId].x == TAMA_NONE) {
           tama[iId].x = iTx;
           tama[iId].y = iTy;
           tama[iId].x1 = iTx1;
           tama[iId].y1 = iTy1;

           if((rand() % 10)==0)
               tama[iId].chn = CHN_POW; /*爆弾*/
           else
               tama[iId].chn = CHN_TAMA; /*敵弾*/

           return;
        }
    }
}

/*--------------------敵弾のスプライト表示*/
void tamaCtr()
{
    int iCnt;
    static int iIdt = 0;
    static int iSpnt = 0;

    for(iCnt=0 ;iCnt<SPN_TAMAMAX ;iCnt++) {

        sprite_set_char(iSpnt ,CFM_CPN | tama[iIdt].chn);
        sprite_set_location(
              iSpnt   ,tama[iIdt].x-(WIDTH_TAMA/2)   ,tama[iIdt].y-
(WIDTH_TAMA/2));

         /*スプライト切り換え*/
        iSpnt = (iSpnt + 1) % SPN_TAMAMAX;
        iIdt = (iIdt + 1) % TAMAMAX;
    }
}

/*--------スプライトx個をまとめて表示*/
/*（スプライト番号 ,x座標 ,y座標, 一辺の長さ ,キャラクター番号 ）*/
void sprite_set_dx(int iSpn ,int iLx ,int iLy ,int iWidth ,int iChn )
{
```

```c
    int iSx,iSy;

    iLx -= (iWidth / 2);   /*描画開始のx座標*/
    iLy -= (iWidth / 2);   /*描画開始のy座標*/
    for(iSy=0 ;iSy<iWidth ;iSy+=CHR_PIX)
    {
        for(iSx=0 ;iSx<iWidth ;iSx+=CHR_PIX)
        {
            sprite_set_char(iSpn ,CFM_CPN | iChn);
            sprite_set_location(iSpn ,iLx+iSx ,iLy+iSy);
            iSpn++;  /*スプライト番号+1*/
            iChn++;  /*キャラクター番号+1*/
        }
    }
}

/*-----------------VBLANK割り込みプログラム*/
void far intr_vblank()
{
    static int iCnt;
    int iIdv=0;

    iCnt = (iCnt + 1) % 4;  /*4回に1回だけ移動*/

    /*------弾移動*/
    for(iIdv=0 ;iIdv<TAMAMAX ;iIdv++) {

        if(tama[iIdv].x == TAMA_NONE)
            continue;

        if(iCnt==0) {
            tama[iIdv].x += tama[iIdv].x1;
            tama[iIdv].y += tama[iIdv].y1;
        }

        if((tama[iIdv].x < 0)||(tama[iIdv].x > LCD_PIXEL_WIDTH)
         ||(tama[iIdv].y < 0)||(tama[iIdv].y > LCD_PIXEL_HEIGHT)) {
            tama[iIdv].x = TAMA_NONE;
            continue;
        }
        if(fnc_abs(tama[iIdv].x ,iMx)< WIDTH_TAMA) {
            if(fnc_abs(tama[iIdv].y ,iMy)< WIDTH_TAMA) {
                switch(tama[iIdv].chn) {
                case CHN_POW:
                    iBomb++;  /*爆弾のプラス*/
                    break;
                case CHN_TAMA:
                    iMhp--;  /*自機のHPマイナス*/
                    break;
                case CHN_BOMB:
                    break;
                }
                tama[iIdv].x = TAMA_NONE;
            }
        }
    }
}

/*----------------------爆発処理--*/
void subTatujin()
{
    int iId;
    int iX1,iY1;

    for(iId=0 ;iId<TAMAMAX ;iId++) {
        tama[iId].chn = CHN_BOMB;  /*キャラクターを爆発に*/
        tama[iId].x = iMx+(rand() % 21)-10;
        tama[iId].y = iMy+(rand() % 21)-10;

        while(1) {
            iX1 = (rand() % 8)-3;
            iY1 = (rand() % 8)-3;
            if((iX1 != 0)||(iY1 != 0))
                break;
        }
        tama[iId].x1 = iX1;
        tama[iId].y1 = iY1;
    }
}

/*---------------------メインプログラム*/
void main () {
    int iKey;
    int iTx,iTy;
    int iCnt;
    intvector_t vectorNew,vectorOld;

    palette_set_color(CPN_MAS, 0x0740);   /*キャラクターパレット番号設定*/
    font_set_colordata(CHN_MIN ,(sizeof(cChrDat) / CHR_BYTE) ,cChrDat);
    sprite_set_range(SPN_MIN ,SPN_MAX );  /*スプライト範囲を設定*/
    lcd_set_color(0x6420, 0xfca8);   /* LCD階調データ初期化 */
    display_control( DCM_SPR );       /*スプライトを表示*/

    subTamaInit();  /*敵弾の初期化*/

    /*----割り込みの設定*/
    vectorNew.callback = (void (near *)())FP_OFF(&intr_vblank);
    vectorNew.cs = _asm_inline("  mov  ax,cs");
    vectorNew.ds = _asm_inline("  mov  ax,ds");
    sys_interrupt_set_hook(SYS_INT_VBLANK, &vectorNew, &vectorOld);


    while (iMhp > 0) {
/*      sys_wait(1);*/  /*ノーウエイトで動作*/

        if((iKey = key_press_check())== KEY_START)
            break;

        /*------自機移動*/
        if((iKey & KEY_DOWN1)&&(iMy < LCD_PIXEL_HEIGHT))
            iMy+=2;

        if((iKey & KEY_UP1)&&(iMy > 0))
            iMy-=2;

        if((iKey & KEY_RIGHT1)&&(iMx < LCD_PIXEL_WIDTH))
            iMx+=2;

        if((iKey & KEY_LEFT1)&&(iMx > 0))
            iMx-=2;

        if(iKey == KEY_B) {
            if(iBomb > 0) {
                iBomb--;
                subTatujin();  /*達人爆弾*/
            }
        }

        /*自機表示*/
        sprite_set_dx(SPN_MY ,iMx ,iMy ,WIDTH_MY ,CHN_MY);

        tamaCtr();  /*敵弾スプライト表示*/

        /*-----弾発射*/
        if(rand() % 2) {
            iTx = (rand() % 2) * LCD_PIXEL_WIDTH;
            iTy = rand() % LCD_PIXEL_HEIGHT;
        }else{
            iTx = rand() %  LCD_PIXEL_WIDTH;
            iTy = (rand() % 2) * LCD_PIXEL_HEIGHT;
        }
        subTamaFire( iTx ,iTy ,fncSgn(iMx-iTx) ,fncSgn(iMy-iTy));
    }

    subTatujin();

    sprite_set_range(SPN_MIN ,SPN_TAMAMAX );   /*スプライト範囲を設定*/
    for(iCnt=0 ;iCnt<100 ;iCnt++)
        tamaCtr();  /*ゲームオーバー表示*/

    sys_interrupt_reset_hook(SYS_INT_VBLANK, &vectorOld);

}
```

携帯電話で動くゲームを作ろう!!

君にもできる「iアプリ」制作



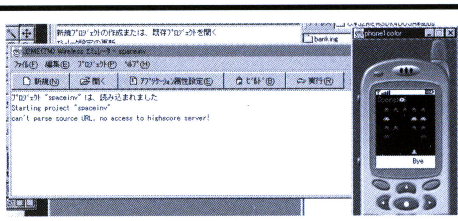
第5弾「インターネット・ランキング機能の追加」

ｉアプリのメリットは「携帯性」と「ネットワーク性」ですが，今回はその「ネットワーク性」にあたる通信機能を使ったゲームを紹介したいと思います。　(松原拓也)

ｉアプリの作りかた

①サンマイクロシステムズのWEBサイト(http://java.sun.com/)からJava2の開発環境「Java 2 SDK Standard Edition 1.3.0」をダウンロードします。インストールすると“C:¥jdk1.3”というディレクトリにJavaコンパイラが展開されます。

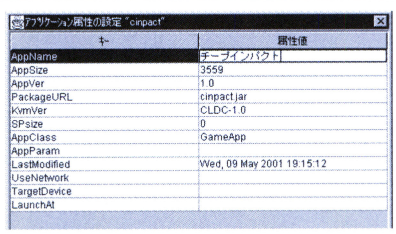
② NTTドコモのWEBサイト(http://www.nttdocomo.co.jp/i/java.html)からｉアプリの開発ツール「J2ME Wireless SDK for the DoJa release2.2(以下，DoJa)」をダウンロードします。ダウンロードには名前の登録が必要です。インストールすると“C:¥J2MEW SDK4 DOJA”フォルダに主要なファイルが展開されます。



③スタートメニューから「J2ME～DoJa」を選択して，「KToolBar」を起動します。

④「新規」ボタンを押して，新規プロジェクト名を入力すると，“C:¥J2MEWSDK4DOJA¥apps”フォルダの下にプロジェクト名のフォルダと，その下に“bin”，“res”，“src”フォルダが作成されます。

⑤“res”フォルダに画像(gif)や音楽ファイル(mld)，“src”フォルダにはソースファイル(java)を格納します。



⑥「アプリケーション属性設定」ボタンを押して，キーの値を入力します(“AppSize”と“Last Modified”は自動的に更新されます)。

AppName = (アプリケーション名)
AppVer = 1.0
AppClass = (最初に実行するクラス名)
SPsize = (スクラッチパッドのサイズ)
KvmVer = CLDC-1.0

⑦「ビルド」ボタンを押すと，コンパイルとプリベリファイが一括して行われ，jarファイルとjamファイルが“bin”フォルダに生成されます。jarファイルは"res"フォルダの中身とクラスファイルがzip形式で圧縮されたものです。

⑧エディタで“C:¥J2MEWSDK4DOJA¥lib¥devices¥phone1color”フォルダの“phone1color.properties”を以下のように書き換えます(値は120×130ピクセル時の設定です)。書き換えたら「編集－設定」をメニュー選択して，「デバイス指定」を“phone1color”にします。




```
  :
screen.x=27
screen.y=65
  :
screen.width=120
screen.height=156
  :
icon.up: 44, 138, off
  :
icon.down: 44, 146, off
  :
screenPaintableRegion.width=120
screenPaintableRegion.height=130
  :
softbutton.0=0,138,36,18, softButton
softbutton.1=90,138,36,18, softButton
  :
```

⑨「実行」ボタンを押すと，携帯電話エミュレータ上でｉアプリが実行されます。

《リスト1》Jam ファイル “ranking.jam”

```
PackageURL = ranking.jar
AppSize = 4224
AppName = ranking
AppVer =
AppParam =
AppClass = GameApp
SPsize = 0
UseNetwork = http  ◀·························· ❶
LastModified = Tue, 10 Jul 2001 18:36:55
KvmVer = CLDC-1.0
```

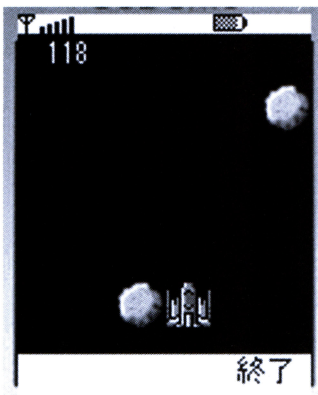
《リスト2》“GameApp.java”

```
/*-------------------------------------------------------------*/
/*「チープインパクト2」 "GameApp.java"  by 松原拓也*/

import com.nttdocomo.util.*;  /*クラスのインポート*/
import com.nttdocomo.ui.*;

public class GameApp extends IApplication
{
    public void start()
    {
        GameScreen screen = new GameScreen(); /*キャンバスの作成*/
        Display.setCurrent(screen);  /*キャンバスのカレント表示*/
    }
}
```

「チープインパクト2」

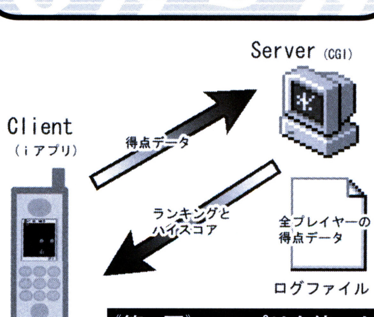


◀インターネット・ランキング機能で全国順位がわかるようになりました

通信機能を使って「チープインパクト(2001年5月号参照)」にインターネット・ランキング機能を追加してみました。

ゲーム終了時に，得点がサーバへ送られ，全国ランキングとハイスコアが集計されます(**第1図**)。その値はｉアプリ側に戻されてダイアログに表示されます(**画面1**)。

ｉアプリ側のプログラム



《第1図》ｉアプリを使ったインターネット・ランキングの構成図

“GameApp.java”(**リスト2**)と“GameScreen.java”(**リスト3**)を入力してビルドします。3枚のGIF画像は前作と同じです。あと，Jamファイル(**リスト1**)にネットワークの使用を設定するのを忘れないようにしましょう。

《リスト3》“GameScreen.java”

```
/* 「チープインパクト2」キャンバス "GameScreen.java" */
import com.nttdocomo.ui.*;  // クラスをインポート
import com.nttdocomo.io.*;
import com.nttdocomo.net.*;
import java.util.*;
import java.io.*;
import java.lang.*;
import javax.microedition.io.*;

public class GameScreen extends Canvas implements Runnable {

    final int TIPW = 20; /* 常数:キャラの一辺のピクセル数 */
    final int MAPW = 6;  /* 常数:画面一辺のチップ数 */
    final int SEQSTART = 0;
    final int SEQDEAD = 1;
    final int SEQEND = 25;
    int iMx,iMy;  /* 主人公座標 */
    int iSpeed;
    Thread myThread;  /* スレッド */
    Image imgMy;      /* イメージ:主人公(宇宙船) */
    Image imgBomb;    /* イメージ:主人公(爆破) */
    Image imgRock;    /* イメージ:隕石 */
    int iSeq;  /*シーケンス*/
    int iScore ,iNando;  /* スコア、難易度 */
    int[][] iMap = new int[MAPW][MAPW];   /* マップデータ */
    Random rnd = new Random();   // 新しい乱数列を作成

    public Image getImg(String sName) { /* イメージ読み込み関数 */
        MediaImage mi = MediaManager.getImage(sName);
        try {
              mi.use(); /* イメージの読み込み */
        } catch (ConnectionException e) {
        }
        return(mi.getImage());
    }

    public void rank(int iScore) {      /*インターネットランキング登録*/     ❷  ❸
      String str = new String("");
      String url = new String("http://www.xxxxxxxxx.co.jp/cgi-bin/rank.cgi");◀····
      try {
          HttpConnection httpconn = (HttpConnection)Connector.open(url,Connector.READ_WRITE,true);
          httpconn.setRequestMethod("POST");  ◀······························ ❹
          httpconn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
          OutputStream out = httpconn.openOutputStream();  ◀·················· ❺
          byte[] bSend;
          String strArg = "score=";

          strArg = strArg + iScore;
          bSend = strArg.getBytes();
          out.write(bSend);      /*サーバに送信*/
          out.close();
          httpconn.connect();  ◀·············································· ❻

          InputStream in = httpconn.openInputStream();  ◀····················· ❼
          ByteArrayOutputStream bytearrayout = new ByteArrayOutputStream();
          byte bRecv[] = new byte[1];

          int len;
          while(true){
              len = in.read(bRecv);
              if(len < 0) break;
              bytearrayout.write(bRecv);   /*データの取得*/
          }
          str = bytearrayout.toString();
```

➡次ページへ

❶「UseNetwork」キーを「http」とすると，通信プロトコルをHTTPに設定されます。そして，ｉアプリのネットワークの使用が許可されます。

❷サーバのURLを格納します。ここで指定した場所にCGIを配置しましょう。

❸Connector.open( )メソッドを呼び出して接続オブジェクトを作成します。

❹HTTPには，サーバから受信しか行わない「GET」と，サーバに送受信を行う「POST」という2種類のモードが存在します。この場合は「POST」を選択します。

❺HttpConnection.openOutputStream( )メソッドを呼び出し，出力ストリームを作成します。OutputStream.write( )メソッドで出力ストリームに書き込み，OutputStream.close( )メソッドでストリームを閉じます。この出力で得点データがサーバに転送されます。

❻HttpConnection.connect()メソッドを使ってリモートオブジェクトを作成します。

❼HttpConnection.openInputStream( )メソッドを使って入力ストリームを作成します。InputStream.read( )メソッドで入力ストリームから読み取り，InputStream.close( )メソッドでストリームを閉じます。この入力で得点の全国順位とハイスコアが得られます。

```
            bytearrayout.close();
            in.close();
            httpconn.close();   ←………………………… ❽
        } catch(IOException e){
            str += e.toString();   /*エラー表示*/
        }

        Dialog dg = new Dialog(Dialog.BUTTON_OK,"結果");
        dg.setText(str);    /*ダイアログの表示*/
        dg.show();
    }

    public void subInit(){   /*ゲームデータのクリア*/
        int iTx,iTy;

        iMx = MAPW/2;   /* 主人公座標 */
        iMy = MAPW-1;
        iSeq=SEQSTART;  /*シーケンス初期化*/
        iScore=0;     /*スコア0点*/
        iNando=0;    /*難易度0*/
        for(iTx=0 ;iTx<MAPW ;iTx++) {
            for(iTy=0 ;iTy<MAPW ;iTy++) {
                iMap[iTx][iTy] = ' ';  /*マップデータのクリア*/
            }
        }
    }

    public GameScreen() {   /* コンストラクタ */
        setSoftLabel(Frame.SOFT_KEY_2, "終了");
        imgMy   = getImg("resource:///ship.gif"); /* GIF画像読み込み */
        imgBomb = getImg("resource:///bomb.gif");
        imgRock = getImg("resource:///rock.gif");

        myThread = new Thread(this); /* スレッド作成 */
        myThread.start();  /* スレッド開始 */

        iSpeed=50;
        subInit();  /*ゲームデータのクリア*/
    }

    public void paint(Graphics g) {   /* ペイントイベント */
        int iGx,iGy,iPos;

        g.lock();  /* 画面のロック */
        g.setColor(g.getColorOfRGB(0, 0, 0));
        g.fillRect(0, 0, getWidth(), getHeight() ); /*宇宙*/

        /* マップの描画 */
        for(iGy=0 ;iGy<MAPW ;iGy++ ) {
            for(iGx=0 ;iGx<MAPW ;iGx++ ) {
                if(iMap[iGx][iGy] == '*' ) {
                    g.drawImage(imgRock ,iGx*TIPW ,iGy*TIPW);
                }
            }
        }

        g.setColor(g.getColorOfRGB(255, 255, 255));
        g.drawString(Integer.toString(iScore), 12, 12);   /*スコア表示*/

        if(iSeq == 0)
            g.drawImage(imgMy ,iMx*TIPW ,iMy*TIPW);   /* 主人公の描画 */
        else
            g.drawImage(imgBomb ,iMx*TIPW ,iMy*TIPW);   /* 主人公爆発 */
        g.unlock(false);  /* ロック解除 */
    }

    public void run() {  /* スレッド実行 */
        int iKey;
        int iTx,iTy;
        char cTmp;

        for(;;) {
            try {
                Thread.sleep(iSpeed);
            } catch(InterruptedException ignore) { }

            iKey=getKeypadState();      /* 主人公の移動 */

            if(iSeq == SEQSTART) { /*シーケンスによる分岐*/

                iScore++;  /*スコアをプラス*/
                if((iScore % 100)==0)
                    iNando++;     /*徐々に難易度をプラス*/

                /*主人公の移動*/
                if((iKey==(1 << Display.KEY_LEFT))&&(iMx > 0)) {
                    iMx--;
                }
                if((iKey==(1 << Display.KEY_RIGHT))&&(iMx < (MAPW-1))) {
                    iMx++;
                }

                /*当たり判定*/
                if( iMap[iMx][iMy] == '*' )
                    iSeq = SEQDEAD;  /*隕石に接触してゲームオーバー*/
            }else{
                iSeq++;  /*爆破中。一定時間見せたら最初に戻る*/
                if(iSeq > SEQEND) {
                    rank(iScore);
                    subInit();  /*ゲームデータのクリア*/
                }
            }

            /*画面スクロール*/
            for(iTy=(MAPW-2) ;iTy>=0 ;iTy-- ) {
                for(iTx=0 ;iTx<MAPW ;iTx++ ) {
                    iMap[iTx][iTy+1] = iMap[iTx][iTy];
                }
            }

            for(iTx=0 ;iTx<MAPW ;iTx++ ) {   /* マップ継ぎ足し */
                if( Math.abs(rnd.nextInt() % 100) > iNando)
                    cTmp=' ';
                else
                    cTmp='*';  /* 隕石出現!! */

                iMap[iTx][0] = cTmp;
            }

            if(iKey==(1 << Display.KEY_SOFT2)) {
                IApplication.getCurrentApp().terminate(); /*終了*/
            }
            repaint(); /*再描画*/
        }
    }
}
```

❽リモートオブジェクトをHttpConnection.close( )メソッドで閉じます。

サーバ側のプログラム

サーバ側はPerlを使ったCGIでプログラムしました。WEBサーバに“rank.cgi”(リスト4)を置き，ファイル属性を実行可能な状態（755）にします。あと“rank.dat”という空のログファイルを作成してファイル属性を書き込み可能な状態にしておきます（666）。

❾接続元の情報「HTTP_USER_AGENT」が「DoCoMo」でないとエラーを表示します。

❿ログファイルを参照して順位を計算します。

⓫ログファイルを比較してハイスコアを算出します。

⓬ログファイルに得点データを追記します。



◀《画面1》ゲーム・オーバー時に、全国順位とハイスコアが表示される

まとめ

こうしてネットワーク機能を使いこなせば，ｉアプリがもっとおもしろくなるのではないでしょうか。

参考URL

今回は中嶋康弘さんの「My iアプリ」というWEBサイトを参考にさせていただきました。

携帯電話でJavaしてみよう！「My iアプリ」
http://homepage2.nifty.com/nicepage/

《リスト4》サーバ用プログラム（Perl）“rank.cgi”

```
#!/usr/local/bin/perl

require './jcode.pl';

#LOGファイル名
$logfile = "./rank.dat";

#最大データ数
$maxdata = 1000;

if ($ENV{'REQUEST_METHOD'} eq "POST") {
    read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
} else {
    $buffer = $ENV{'QUERY_STRING'};
}

if(index($ENV{'HTTP_USER_AGENT'},"DoCoMo") == -1){
     &error('携帯からアクセスしてください'); ◀··········❾
}

if($buffer ne "") {
    &addlog;
}

if (!open(DB,"$logfile")) { &error('ファイルを読み出せず'); }
@lines = <DB>;
close(DB);

print "Content-type: text/html¥n¥n";

$win=1;
$max=0;
foreach $line (@lines) {
    if ($line =~ /(.*)/) {
        if($1 ne "") {
            if($write < $1) {
                $win = $win + 1; ◀··········❿
            }
            if($1 > $max) {
                $max = $1; ◀··········⓫
            }
        }
    }
}
print "Score $write¥n";
print "Ranking $win位¥n";
print "Hi Score $max¥n";
print "¥n¥n";
exit;

sub addlog {
    @pairs = split(/&/,$buffer);
    foreach $pair (@pairs) {
        ($name, $value) = split(/=/, $pair);
        $value =~ tr/+/ /;
        $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
        &jcode'convert(*value,'sjis');

        $value =~ s/</&lt;/g;
        $value =~ s/>/&gt;/g;

        if ($name ne "write") { $FORM{$name} = $value; }
    }

    if ( $FORM{'score'} ne "") {
        $write = $FORM{'score'};
    }else{
        &error('引数なし');
    }

    if (!open(LOG,"$logfile")) { &error('ファイル読み込めず'); }
    @lines=<LOG>;
    close(LOG);

    $axs = @lines;
    $maxdata2 = ($maxdata - 1);

    if($axs >= $maxdata) {
        if (!open(LOG,">$logfile")) { &error('ファイル記録できず'); }
        foreach $axs (1..$maxdata2) { print LOG $lines[$axs]; }
    }else {
        if (!open(LOG,">>$logfile")) { &error('ファイル記録できず'); }
    }
    print LOG "$write¥n"; ◀··········⓬
    close(LOG);
}

sub error {
    print "Content-type: text/html¥n¥n";
    print "$_[0]¥n¥n";
    exit;
}

# end of script.
```

超 初心者向けJava入門 魁!! Dr.JのJAVA-WALK ジャバウォック



第3回 もうすこし，とりあえずプログラムを実行しよう！

お詫びと修正！？

やぁ，諸君！Dr.Dの従兄弟こと，Dr.J（ドクトルジェイ）である！そろそろみんなワシのことを覚えてくれたかのぅ？それにしても今年の夏は暑い！もうコヨミの上では9月だというのに，まるで毎日が温泉の脱衣所のようである！…って，何のたとえかさっぱしわからんが…要するに年寄りは温泉が好きだっちゅうコトじゃの。

ところで，前号の記事で，画面でのコマンド投入イメージ(枠線で囲んだ部分じゃ）で，半角「¥」が半角「＼」と表示されておった。これは，印刷の文字フォントに欧文フォント(漢字やひらがなの日本語文字が入ってないやつだな)が使われてしまったために起きた過ちじゃ。

日本語環境では半角の¥記号が，欧文フォントでは，同じ文字コード番号が割り当ててある＼記号にすり替わってしまったというわけじゃ。ちなみに，キーボードによっては(とくに自作マシンの場合)，¥記号の他に＼記号のキーもキーボード上にあったりする。諸君の想像どおりに，日本語フォントの場合は＼記号を押しても¥記号として表示されるものであったりするし，日本語サポートの入っていない環境ならば記事のとおりに＼記号で表示される。

すなわち，必ずしも前回の文字ちがいは間違いとも言いきれんのじゃが，読者諸君の誤解のないように理解してくれたまえ。

それにしても，ディレクトリの区切り文字なんてUNIXだと"／"なんじゃよな。間違いやすいったらありゃあせん…。

前回からの続き ... クラスの名前を変えてみよう

というわけで，前回最後に出したナゾを解いてみよう。

ファイル　Helloworld.java

```
public class Helloworld {
    public static void main(String args[]) {
        System.out.println("Hello World");
    }
}
```

このソース・ファイル名をソース・プログラム中の"class"に続く"Helloworld"と合わせておる。これをちがえるとどうなるか，ということじゃったが，どうかな，試してみてくれたかな？

では，まずはこんなふうにソース・プログラムを変更してみよう。

public class Helloworld {

↓

public class HelloBASIC {

これをコンパイルすると ...

```
C:¥javawalk>javac Helloworld.java
Helloworld.java:1: クラス HelloBASIC は public であり，
ファイル HelloBASIC.java で宣言しなければなりません。
public class HelloBASIC {
       ^
エラー 1 個
```

クラス？クラスって学級とか階級とか？ちがう！この場合の"class"とはカタマリとしての意味なのだ！そう，じつはJavaではプログラムの単位を「クラス」と呼んでいるのである！この場合は一つのソース・プログラムが一つのクラスに対応しており，これが「public」という言葉で意味づけをされて（「宣言」というぞ）おる。"public"とは英語で「公開」とか「公衆」といった意味だから，このプログラムを外から見たときに「公開されて見える」のはHelloBASICというクラスなのであった。Javaではお約束として，publicで宣言したクラスは公開されるものであるから，外部から見えるファイル名に使う，ということになっておるのだ。わかりやすいじゃろ？

一つのソース・ファイルの中に，二つとか三つとか，複数のクラスがあってもよいのじゃが，publicをつけられるクラスは一つだけなのは，そのためなのだ。

以上をチョットまとめてみた。お約束として覚えておいてくれたまえ。



●Javaのプログラムのかたまりの一つを「クラス(class)」という。

●publicで修飾されたクラスはソース・ファイルの名前になる。

●一つのソースファイルにpublicのクラスは一つだけ。

「クラス」はJAVAでプログラムを書くうえで，最も大事なコトの一つだが，なかなか理解しづらいかもしれん。一度にいろいろ，ちゃんと理解しながら覚えるなんてのはワシの想定しておる「超初心者」な読者諸君には大変だ！少しずつ覚えていこうではないか！これよりくわしくは，また今度じゃ。

エラー・メッセージはこう読むのだ

ところで！ついでだから，さっきのエラー・メッセージの読みかたを覚えておこう。

よほどのプログラムの達人でもない限り，エラーは必ず起きる！エラー・メッセージをなくすことがすなわちプログラム完成への道のりであるのだから，エラーが表示されたときに「あー，なんかエラーが出てるなぁ…よくわかんないからテキトーに直しちゃえ」なんてのは，言語同断！エラー・メッセージの内容を正しく理解し，適当な（適当とテキトーは大きく意味がちがうのだ！）修正ができるようになろうではないか！

・最初の

Helloworld.java

はエラーのあったファイル名

・つぎのコロン

:

は区切り文字

・そのつぎの

1

はエラーのあった行番号

・つぎのコロン

:

も区切り文字



・つぎのメッセージ

クラス HelloBASIC は public であり，ファイル HelloBASIC.java で宣言しなければなりません。

はエラーの内容

・つぎは2行で一つの意味で，

```
public class HelloBASIC {
         ^
```

エラーのあったプログラムの行の記述と，問題を検出した「場所」を

"^" の文字の上の場所で示す

・最後の

エラー 1 個

はエラーの数

というわけで，結構JAVAコンパイラのエラー・メッセージはわかりやすいように表示されておる。ちゃんと日本語なので，読みやすくてありがたいのう！

つぎは数値計算をしてみよう

さて，さっきは文字を表示できたので，今度は数字～数値計算の結果を表示してみよう。まずは例によってソース・プログラムを入力してみてくれたまえ。

Toshizan.java

```
public class Toshizan {
    public static void main(String args[]) {
        int year_tsugumi, year_kage, year_doctor;
        year_tsugumi=20;
        year_kage=35;
        year_doctor=year_tsugumi+year_kage+10;
        System.out.println("Dr.D - " + year_doctor);
    }
}
```

で，こいつを実行させると…

```
C:¥javawalk>javac Toshizan.java ◀………… コンパイル
C:¥javawalk>java Toshizan ◀………… 実行
Dr.D - 65
```

おお，うまくいったようじゃな。

今回はこんなコトをしておるぞ。ポイントになる行を見ていこう。

int year_tsugumi, year_kage, year_doctor;

int 整数の宣言

year_tsugumi,year_kage,year_doctor; 整数で宣言された変数

year_tsugumi=20;

変数への代入。つぐ美ちゃん20歳！

year_kage=35;

おなじく変数への代入。影さん35歳！（ホントか！？）

year_doctor=year_tsugumi+year_kage+10;

変数への代入。ドクターのトシはつぐ美ちゃん＋影さんだった！

System.out.println("Dr.D - " + year_doctor);

文字列" "につなげて表示したい場合は ＋ をつけて書けばよいのだ。

まぁ，単なる足し算をするだけじゃから，こんなもんじゃろう。理解できたかな！？

それでは，次回までの宿題じゃ。

「1から100までの数を足して，答えを表示する」

じゃ！難しいかのう？では，ヒント…というか，実質的に答えみたいなもんじゃが…「繰り返し文を使う」！のじゃ！

```
for (初期値設定;条件判定;条件更新){
    繰り返される処理
}
```

どうかな？なんとなくわかるかな？

ちなみに，1から10までならこう書くのじゃぞ。

```
for (i=1;i<=10;i++){
    繰り返し処理
}
```

では，残暑厳しい折じゃが，夏バテ対策にJavaのレッスンじゃな！心頭滅却すれば火もまた涼し…あちちち。ではまた来月，To Be Continued じゃ!!

Catlikeの もっと身近にプログラミング！

第5話 シューティング・ゲーム編 その1 フレームスキップ付きメインループを考える

プログラムを組むとき，最初にどこから手をつけますか？　今回はほとんどの人が最初に作るであろう，メインループのお話。意外に見落としがちな部分なんですよね。

筆者紹介

catlikeホームページ
http://catlike.marchen.to/
いろいろ遅れ気味のcatlike企画各種ですが,ようやく公開のめどが立ってきました。「いつか,何処かで。」これからどうぞかわいがってやってくださいませ。

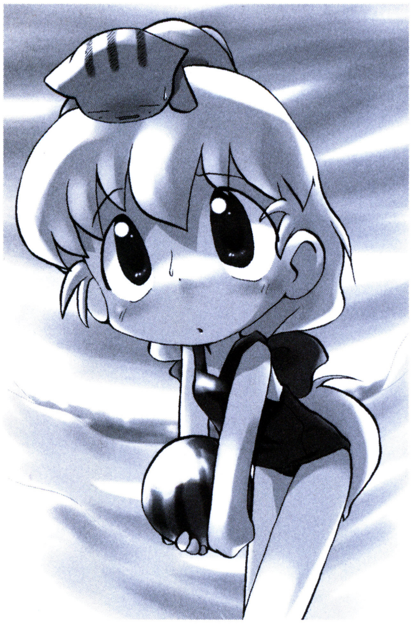
文:まかべひろし
Dreamcastのカラオケセット買いました。ADSLと合わせて,常時接続でセガカラを楽しんでます。できることが,いろいろ増えてくると,もう生産しない機械なのは惜しいなあと強く思うところです。もう1台買っておこうかなぁ…。

絵:桜部のとす
紙を変えたらGペンが使いやすくなりました〜。やっぱり相性とかってあるモノなんですね。あ,あとメモリの値段がどんどん下がってますね〜。メインマシン,とうとう1G積みましたよ,わ〜い。

そろそろ文化祭の季節？

夏休みも終わって，暑い日々が続くけれど，体調大丈夫ですかー？　かき氷食べ過ぎてませんか？　やっぱりかき氷はイチゴミルクが…と前回をひきずるのはやめにして。いやはや，今年の夏は超絶的な猛暑が続いていて，身体に何かねっとりしたものが常にへばりついているような気がします。

そんな中，学生さんはそろそろ文化祭，という学校が多いようですね。展示品なり演劇なり模擬店なり，作品作りに没頭していると，この暑さもパワーに変わる。



そんな気がしたものです。いやまあ過去形なのはボクがもう学生さんじゃなくなってしまったからなだけ。作品作りの情熱をパワーに変えているのは，お仕事でゲームを作っていても変わらないものです。

今年の文化祭は，みなさんどんな作品を作っていますか？　もしゲームを展示する学校がありましたら，ぜひ招待してくださいませ。遊べるとあらば，全国どこへだって飛んでやる！　という意気込み十分な状態です。いえ，決して仕事で詰まってるから旅行したい，なんて思ってるわけじゃありませんよ，ホント。

今回はちょっと視点を変えて

相変わらず，前書きはまったく内容に関係なく，今回のお話にずばばんと入っていったりします。また今回は新しいお話のスタートですね。ゲームのジャンルはいろいろあって，やっぱり新しいお話をはじめるときは，どれにしようか迷ってしまっている状態。うんうん思考を巡らして，今回は，シューティング・ゲームを取り上げてみたいと思います。しかし，シューティング・ゲーム編と言ったところで，じつは内容，計画を立てていったらなんとまぁ，アクション・ゲーム編とほとんど変わりがないじゃないですか。当たり判定やって，移動のルーチンやって，これじゃホントにアクション・ゲーム編の焼きなおしとなってしまう…。

と，ここでちょっと視点変換。今までのように「ポイントに注目して解説する」形ではなく，メインループをいかにして書けばいいかと思ったわけです。実際それで計画を練ってみたら，うん，これはいける，ということでシューティング・ゲーム編，今回のお話は「メインループ」の書きかたに関して，考えていきたいと思うのです。というわけで「シューティング・ゲームで考える，メインループの作りかた」。

メインループはシューティング・ゲームだけでなく，すべてのゲームの基本。どんなゲームだって，このメインループははずすことはできません。プログラムって，アイデアが頭の中にあっても，実際にその部分を組むまでには初期化など

の下準備が必ず必要なもの。汎用性の高いメインループを作っておけば，すぐにゲームの核の部分に手をつけることができるのです。

ゲーム全般に使えるメインループ

まず，ゲームにおけるメインループとはいったいなんなのかを考えてみましょう。ループというのは大体において，1ステップ分を繰り返す処理のことを言います。1ステップはどこからどこまでの処理に値するのかというのは，場合と状況によってちがいますけど，メインという名前がついているぐらいですから，大きなくくりになることは間違いありません。

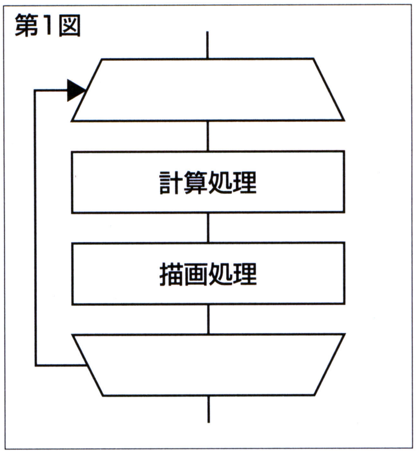
主人公の動き？　いいえ，それだけれはメインになりえません。オープニングからエンディングまでの一連の動作？　それはプログラムにおいてループというには，あまりにも大きすぎます。たしかにループかもしれませんが，それはプログラムを組む土台には到底なりえません。

では，プログラムにおけるゲームの実際の動きというのを考えてみましょう。まず，主人公や敵の動きを計算します。シューティング・ゲームなら当たり判定や得点計算もその中に入ります。すべての計算処理が終わったら，計算結果を視覚的に表現するために，画面に描画します。そしてまた再び，主人公や敵の動きに戻ります。

おや。見事にループしているじゃありませんか。計算，処理，描画，また計算…と，これがゲーム・プログラムにおけるメインループとなるのです。ゲームというアプリケーションが，他のアプリケーションと一番ちがう部分，それは画像描画が常に行われ続けることです。すべての処理は，1フレームという秒間に何10枚も表示されるうちの1枚を作り出すためのプロセスと言えるでしょう。やっぱりゲーム作るときは，画面描画をいかにかっこよく，かわいくするかに力を入れちゃいますからね。

そんなわけで，メインループは，描画処理を区切りとしたプログラムとなります。これだけを考えると，**第1図**のような感じですね。

第1図



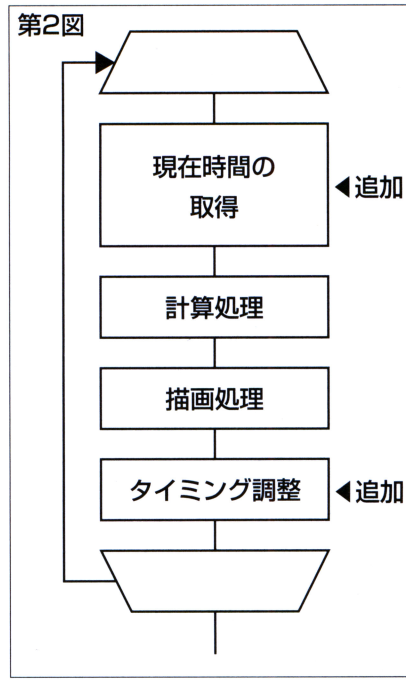
当たり前のように見えますが，このメインループには「時代」という大いなる欠点があります。今のパソコン環境は，何度も繰り返しているように，さまざまな環境があります。同じパソコンだって，同時に立ち上げている他のソフトによっては処理できる能力が変わってきてしまう状況なのです。ゲームというのはタイミングが非常にシビアなプログラムなので，できる限りどのマシンでも同じようなタイミングで動作するように作らなくてはいけません。

ところで，フレーム数という言葉を聞いたことがあるでしょうか。よく3D系ゲームで言われる言葉ですが，秒間30フレームとか60フレームとかいう使いかたをします。要は，何分の1秒で1回描画ができるかということを表現しています。秒間60フレームなら，1枚描画するのに1/60秒を要するわけですね。

最近のCPUは速いですから，簡単な計算処理くらいは1/60秒以内にやってのけてしまうでしょう。しかし，描画処理がかなりネックになってきます。ここはまず，描画処理を含めた1ループ内での処理がすべて1/60秒で行われるという前提で話を進めていきます。

第1図のプログラムではタイミング調整をするような部分は何もありませんから，CPUは力の限り処理を続けていきます。その結果，1ループ内の処理が1/60秒以内で収まった場合，1/60秒を待たずにそのままつぎの処理へ移ってしまいます。結果，想定したスピードよりも速い動作を行ってしまうのです。このためには，タイミング調整を行わなければなりません。

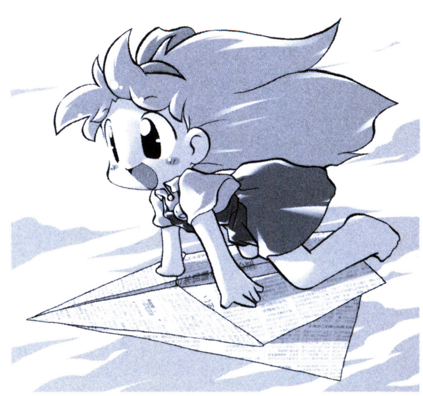
第2図



では，1/60秒というタイミング調整はどのように行えばいいでしょうか。**第2図**を見てください。「現在時間の取得」と「タイミング調整」という二つの項目が追加されています。処理が始まる前に時間を取得しておき，すべての処理が終わってからタイミング調整部で，処理開始時から1/60秒が経過するまで待てばよい，という理屈です。

さてここで，ゲームを動作させる環境で，調べてほしいことがあります。その環境，つまりWindowsとかPlayStation2 BASICとかに，1/60秒より細かい精度で動作するタイマは搭載されているでしょうか。Windowsの場合，標準でGetTickCount（）というAPIが用意されており，これは1/1000秒単位でカウントアップされるタイマを読み出すことができます。このようなタイマが用意されていれば話は簡単。

まず，処理開始前にタイマの値を読み



出しておき，変数に保存します。そして，タイミング調整部にきたら，再びタイマの値を読み出し，保存しておいた値に1/60秒分のカウント，たとえば1/1000秒の精度を持つタイマなら1000/60の値を加算してやり，予定時間が経過しきるまで待ってやるのです。

```
nStartCnt をループ開始時の値
GetTickCount（）を1/1000秒カウンタとする場合，
以下の命令で時間待ちを行う

while（nStartCnt＋（1000 / 60）＞GetTickCnt（））;
```

また，環境によってはこのようなタイマは用意されていないかもしれません。ゲーム機の場合は「モニタのVSyncを待つ」という命令があり，これをタイマの代わりに用います。VSyncについては説明を省略しますが，家庭用テレビを使っている場合，1VSyncはほぼ1/60秒という時間であり，タイミング調整部にVSync待ちの命令を入れることで，1/60秒が経過するまで待つという目的を果たすことができます。

ちなみに，1000/60を計算すると16.666……となり，割り切ることができません。C言語などでint型宣言をした場合，小数点以下の値は勝手に切り捨てられてしまい，16という値で扱うことになると思います。正確な1/60秒は計れないじゃないか，と思うかもしれませんが，これはあくまでゲームの動作速度を決定するための値であり，それほど1/60秒という値にこだわる必要はなかったりします。大事なのは，この値をベースに動作速度を考えること。ゲームをプレイする人は，このソフトがどれくらいのフレーム数で動くのか，などと気にする必要はないのです。

このように「時間待ち」を入れてやることで，想定した速度よりも速く動作することはなくなります。

フレームスキップを入れなきゃね

しかし，これまでの説明はあくまで「すべての処理が規定時間内で終わること」が絶対条件でした。ではもし，規定時間内で処理が終わらなかった場合はどうなるでしょうか。Windowsなどでは組みかたによっては，1/60秒で描画を含めたすべての処理が終わらないことなど当たり前だったりします。それこそマシン環境いかんでは，1/60秒で描画できるマシンもあれば，描画に1秒間かかってしまう環境だってありえるわけです。単純に考えて，この二者の動作速度には60倍もの差が出てきてしまっています。

さて，通常のゲーム制作において，一番処理時間がかかる部分，ボトルネックと呼ばれる部分はたいてい描画部分につきます。メモリ上の計算処理なら，最近のCPUには朝飯前といった感じです。しかしひとたび描画部分で半透明処理を行ったり，回転処理を行ったり…と負荷をかけていけば，なかなかの重労働であることは一目瞭然。ハードウエアで行うのが当たり前の3D処理をCPUに肩代わりさせたりなどしたら，秒間1フレームというのは冗談ではなくなってしまいます。

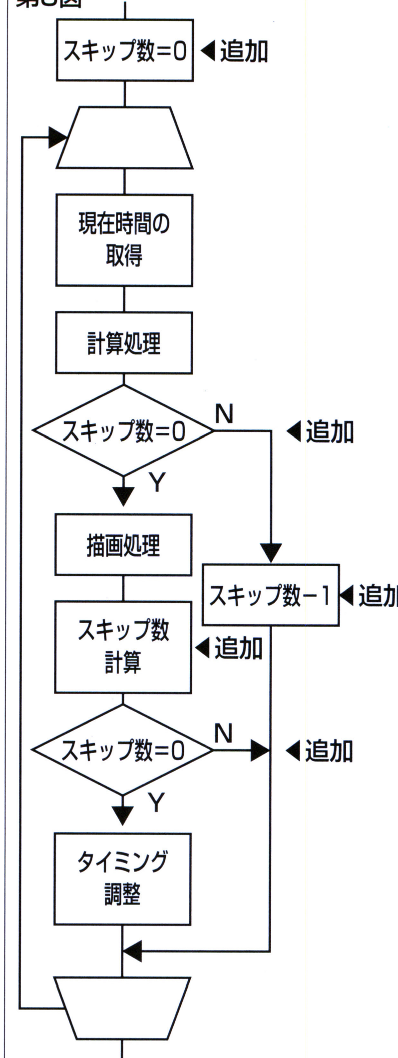
そこで，内部計算は毎回行うけれども，描画処理自体を飛ばしてしまうことで，どうにかゲーム自体の動作速度は変更しない，という苦肉の策を取るのが一般的になっています。フレームスキップ，という用語も当たり前のように使われているほど。1/60秒に毎回描画しては遅れてしまう環境でも，描画を3回に1度，1/20秒ごとの描画にすればなんとかついていけるかもしれません。CPUの作業数が少なくなっているのは目に見えています。

考えかたとしては，「1ループで超過してしまった時間を，つぎのループ以降で取り戻す」という方法を取ります。完全無欠の正確なフレームスキップとは異なりますが，実装方法が簡単なのが特徴です。**第3図**がその流れ図ですが，ループの前の初期化部分で，スキップ数という変数を追加しています。

第3図にある「スキップ数計算」が，今回の追加のメインでしょう。これは，**第2図**で追加のタイミング調整で使ったカウンタで行うことができます。

現在時間から，本来終わるべき時間を引いてやった超過時間を，フレームの基準値で割ってやればいいのです。

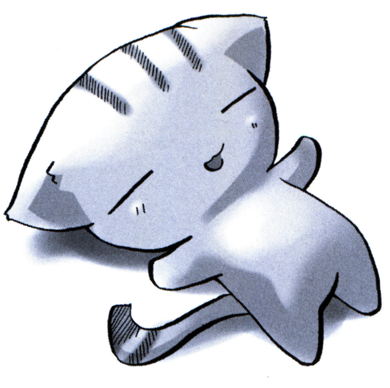
第3図




```
nStartCnt をループ開始時の値
GetTickCount（）を1/1000秒カウンタとする場合，
以下の命令で nSkipCnt にスキップ数を得ることができる

nNowCnt＝GetTickCnt（）;
if（nStartCnt＋（1000 / 60）＜nNowCnt）{
    nSkipCnt＝（nNowCnt−（nStartCnt＋（1000 / 60）））/（1000 / 60）＋1;
} else {
    nSkipCnt＝0;
}
```

このサンプルで，nSkipCntの最後に＋1しているのは，少しでも超過した場合はフレームスキップを行うための処置。こうしないと，「なんか全体的にもったりしてるんだよなぁ」という状況になっ



てしまうわけです。

この方法でフレームスキップを実装する場合，描画を抜かした1ループの処理が1/60秒以内に終わるということが前提条件になります。もし，時間内に処理が終わらないような環境も動作保証に含める場合，ベース時間を1/30秒にしてみるとか，計算処理自体を簡略化するなどの工夫が必要になってきます。どうにもならない場合，動作保証環境をを引き上げるという処置を取ることになります。まぁ，CPUは高速化の一途をたどっていますから，計算処理程度ではそうそうビクともしないことでしょう。

ジャンルによってはオプションを

さてさて，前述のように組めばフレームスキップも搭載し，どんな環境でも，描画精度のちがいはあるものの，同じようなタイミングで動作するゲームを作成することができます。ここまで作っておけば，いざ新しいゲームのプログラムをしようと思ったとき，すでにある程度のコードが書かれたベースがあるわけで，取っ掛かりが非常に気楽という大きな大きな利点があります。

しかし，フレームスキップということは，中間の情報が欠落してしまうということに他ならず，アドベンチャー・ゲームやロールプレイング・ゲームのように「キー待ちが入って，自分の判断で進行を進める」ゲームならいざ知らず，今回のテーマであるシューティング・ゲームや，アクション・ゲームのように，「プログラムが強制的に進行を進める」ゲームの場合は，情報の欠落が大きな痛手になりかねません。考えてもみてください。フレームスキップによって，目前の敵がミサイルを発射したフレームが欠落し，つぎの瞬間には自機がやられてしまっている状況。これではゲームとしてとても遊べるものではなくなってしまいます。

しかし，プログラムでどんなにがんばっても，マシンの基本性能を上げることはできません。プログラム処理の高速化にも限界というものがあります。

そこで編み出された策として，「処理落ち」というものがあります。シューティング・ゲームで大量の弾が発射されると，動作が明らかに遅くなって，弾を避けやすくなるというものがあるでしょう。アレです。本来スキップすべきフレームを表示する代わりに，動作速度を遅くしてなんとか処理してやるという方法論です。

また，シューティング・ゲームで処理落ちが起きるほとんどの場合は，敵機や弾が大量に発生したことによるものなので，ゲーム自体の難易度を調整する役目も持っています。山ほどの弾をすいすいすり抜ける快感は，何物にも変え難いものがあります。

処理落ちを実装する方法は，フレームスキップを搭載しているプログラムなら意外に簡単です。たとえば1/20秒以上フレームスキップが起きた場合，処理落ちとして処理する場合，スキップ数計算を以下のようにします。

```
nStartCnt をループ開始時の値
GetTickCount（）を1/1000秒カウンタとする場合，
以下の命令で nSkipCnt にスキップ数を得ることができる。
1/20秒以上は処理落ちと判断し，必要以上のスキップをしない。

nNowCnt＝GetTickCnt（）;
if（nStartCnt＋（1000 / 60）＜nNowCnt）{
    nSkipCnt＝（nNowCnt－（nStartCnt＋（1000 / 60）））/（1000 / 60）＋1;
} else {
    nSkipCnt＝0;
}
if（nSkipCnt＞（60 / 20））{
    nSkipCnt＝（60 / 20）;

}
```

追加したのは最後の部分。もしフレームスキップによって1/20秒以上飛ばされるのなら，スキップ数を規制してやる，という内容です。こうすることにより1/20秒まではスムーズに動き，それ以上遅れるようなら，内部時間で1/20秒に1度は強制的に描画してやりましょう，という形になます。結果的に，規制された分のフレームが処理落ちという形で表示されるわけです。

最近のゲームでは，『イース1・2完全版』が，この処理落ちをオプションで選べるようになっていましたね。最高のCPUと最低レベルのCPUの差がどんどん大きくなっていく現在，より多くの動作環境での楽しいゲームを保証するには，こういったオプションの設定はこれからは不可欠になっていくのかもしれません。

次回も，今回と同じようにメインループのお話。今回は図でも完全に無視した「計算処理」の部分を突っ込んで取り上げたいと思います。

catlike最新情報

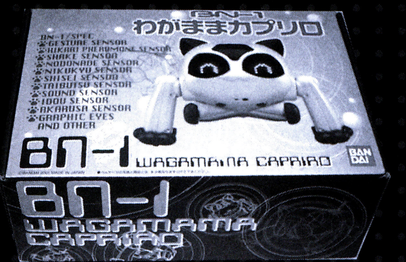
『いつか，何処かで。』のWebページを公開してから，多くの反響をいただいております。まだWebページしか公開できていないのに，です。いやはやすいません。どうにもこうにも時間が取れなく，公開できる状況にまで至っていません。しかしそろそろ，という状況にはなってきました。1日も早い公開を目指してがんばっていますので，どうぞよろしくお願いします～。

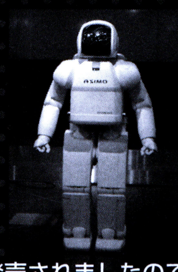


▲『いつか，何処かで。』公式ページで情報公開中！

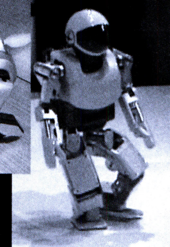
連載 第5回

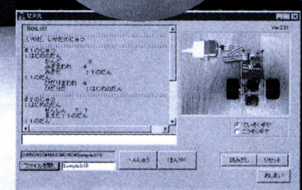
21世紀の遊びはコレ！ パソコンでロボットをプログラミングしよう!! ロボ道場













遂にBN-1のプログラム・キットが発売されましたので，BN-1本体と合わせて紹介します！　（松原拓也）

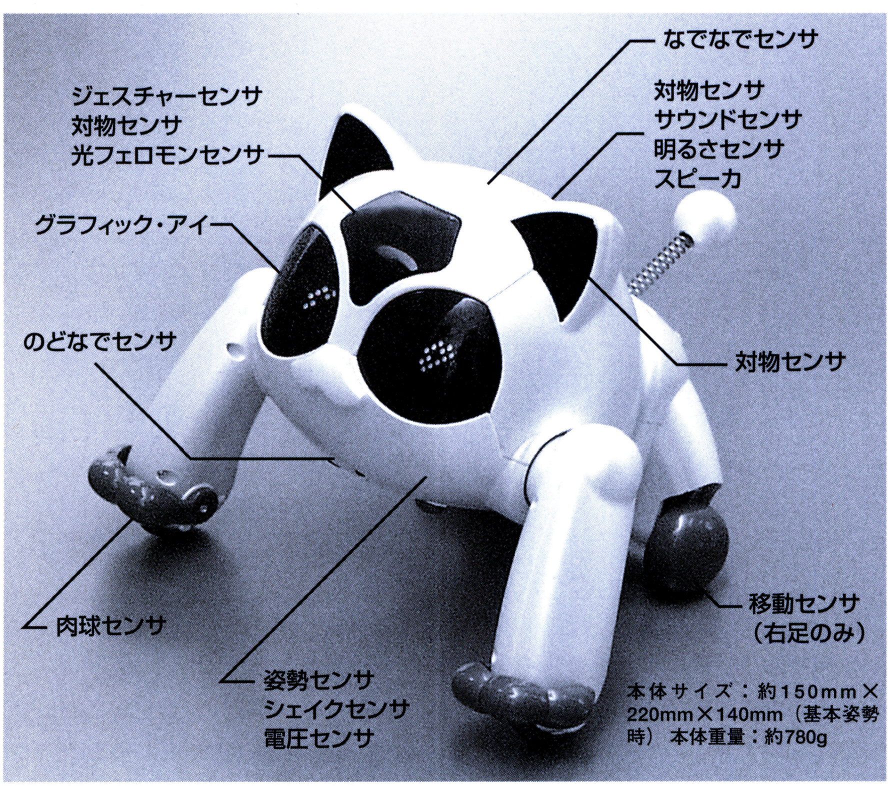
BN-1 わがままカプリロ

「BN-1 わがままカプリロ」
バンダイ　価格50,000円（税別）
※「BN-1 わがままカプリロ」はLaLaBitMarketやツクモロボコンマガジン館などで購入可能です。

BN-1わがままカプリロ
オフィシャルホームページ
http://www.bn-1.channel.or.jp/

ロボ道場 A.I.搭載の猫型ロボット

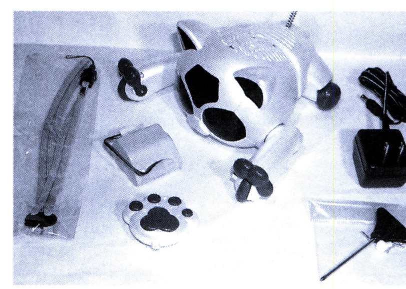
「BN-1 わがままカプリロ」は人工知能（A.I.）を搭載した本格的なロボットです。7種類13名称ある内蔵センサのうちの一つ，ジェスチャーセンサは飼い主の手振りを認識することができます。そして外見では，大きな両腕，下半身の車輪，両目の大きなグラフィック・アイ，カワイイ猫型の顔が特徴です。



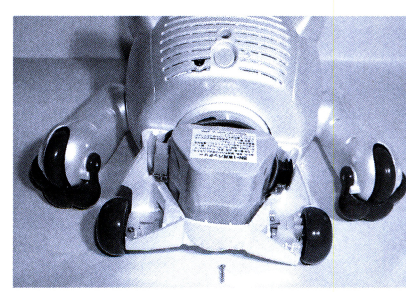
本体サイズ：約150mm×220mm×140mm（基本姿勢時）　本体重量：約780g

ロボ道場 最初に

BN-1を購入したら最初にバッテリを組み込んで，初期化を行います。



▲箱の中身はこのとおり（この他にマニュアルが付属）



▲BN-1の下半身をドライバで開けて，バッテリを組み込む

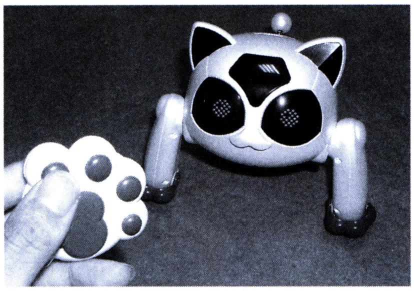


▲充電器でバッテリをチャージさせ，一旦メモリをリセット。そして，BN-1の電源を入れたら完了

ロボ道場 BN-1を遊んでみる

BN-1の成長パターンはあらかじめ決められていません。このため，飼い主の接しかたで行動や性格が幅広く変化します。行動パターンは「お手」や「でんぐりがえし」や「ダンス」など約100パターンほどあります。

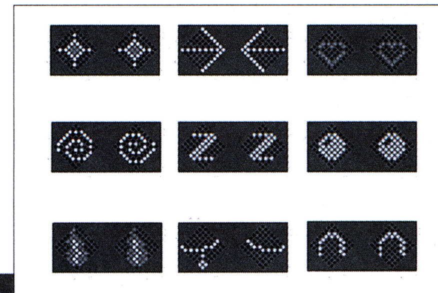
通常のモード以外にもBN-1には「ゲームモード」「リモコンモード」「モーションエディットモード」というモードがあります。



▲「IDスイング」で飼い主の認識やリモコン操作が可能



▲下半身を持ち上げて踊る荒業「お尻ダンス」。腰の回転角は320度と広範囲



▲両目に付けられたグラフィック・アイの表現数は250パターン

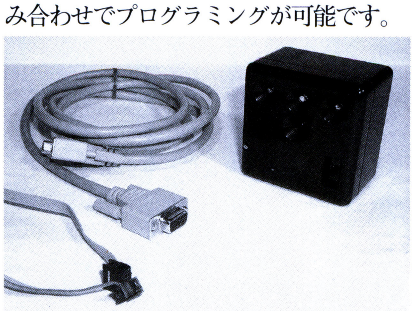


▲なでると「なでなでセンサ」が反応。目がニコニコになったり、ハートになったりと非常にカワイイ

ロボ道場 「BN-1わがままカプリロ プログラミング・キットver.1.0」 価格12,000円（税別） 対応機種Windows95/98/Me/NT4.0

このキットに付属する「A.I.TRAINER」というパソコン用のアプリケーション・ソフトには，初心者向けのツール「A.I.TRAINER STANDARD」，上級者向けのツール「A.I.TRAINER PRO」，モーションエディット用ツール「A.I.TRAINER MOTION」の三つの機能が用意されています。

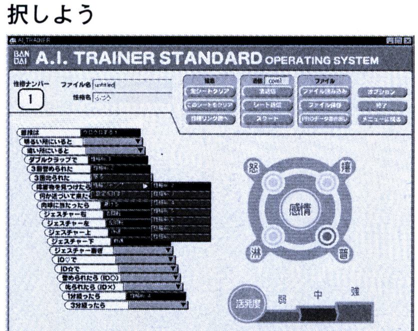
「A.I.TRAINER STANDARD」では「どうだったら」「どうする」という簡単な組み合わせでプログラミングが可能です。



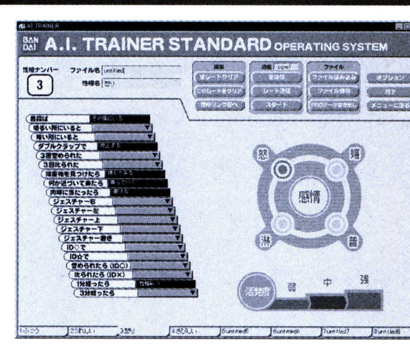
▲「プログラム・キット」に付属する専用赤外線インターフェース・ユニットとケーブル類（この他に電池，説明書，CD-ROMが付属します）



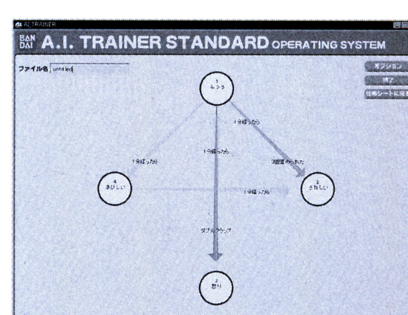
▲「A.I. TRAINER」の画面。初心者はまず「A.I.TRAINER STANDARD」を選択しよう



▲パネルを切り換えれば，BN-1の感情を変化させることができる



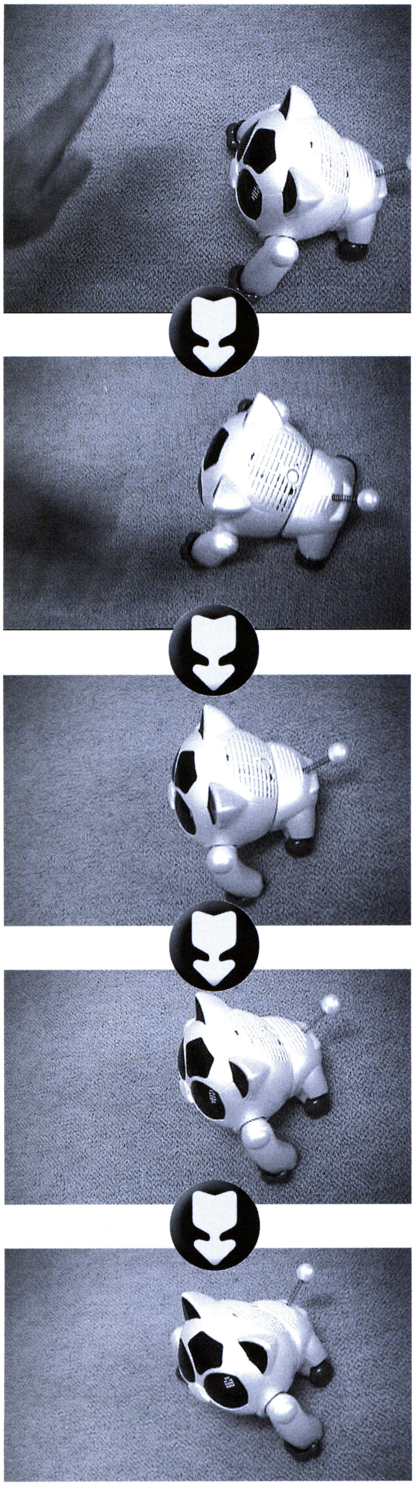
▲怒ったときや，寂しいとき，うれしいとき，普通のときの反応を作成



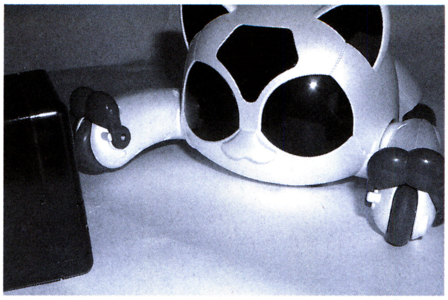
▲パネル（感情）の移り変わりは状態遷移図となって自動的に表示される

ロボ道場 プログラム実行！

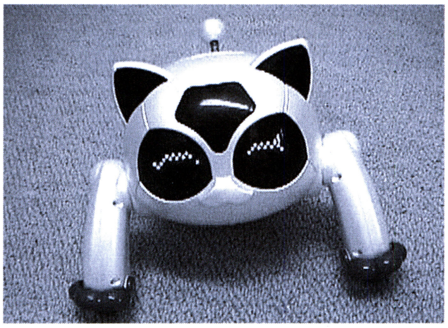
専用赤外線インターフェース・ユニットとパソコンを付属のシリアル・ケーブル（Dsub9ピン）で接続します。「転送」ボタンでプログラムをパソコンからBN-1に送信したら，「実行」ボタンでプログラムを走らせます。



▲プログラムどおりに手のジェスチャーを認識したようす



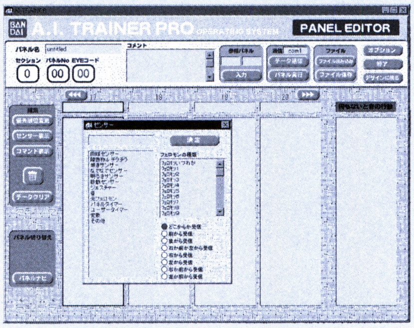
▲赤外線インターフェース・ユニットを使ってBN-1をプログラム中



▲プログラム実行中。「うれしい」とBN-1のグラフィック・アイがうれしそうに表示！

ロボ道場「A.I.TRAINER PRO」

「A.I.TRAINER PRO」は「A.I. TRAINER STANDARD」よりも複雑で，凝ったプログラミング可能な上級者向けのツールです。



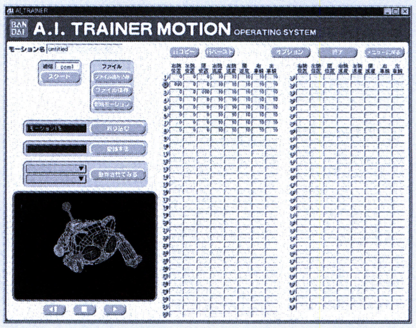
▲センサの選択画面では光フェロモンなどの設定も可能



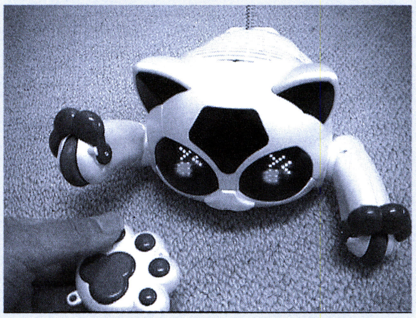
▲コマンドの選択画面。BN-1の持ち芸の多さに改めて驚き（写真は大技の「オヨヨダンス」）！

ロボ道場 モーションエディット

「A.I.TRAINER MOTION」では，モーションエディット（BN-1の各サーボや車輪の制御を編集する機能）をパソコン上で行うことができます。



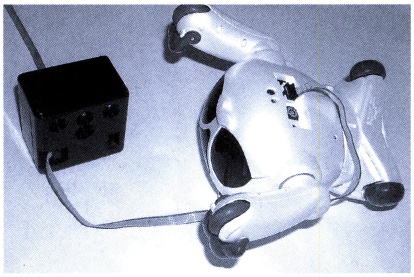
▲「A.I.TRAINER MOTION」の画面。右側が各関節の角度で，左側がプレビュー画面



▲これはBN-1本体だけでできるモーションエディット・モード。作成したモーションはPCでも取り込める

ロボ道場 ロボットシステムの更新も可能！

プログラムキットにはBN-1のフラッシュROMを書き換える機能があります。本記事では使いませんでしたが，将来，ロボットシステムの更新時に威力を発揮するかもしれません。



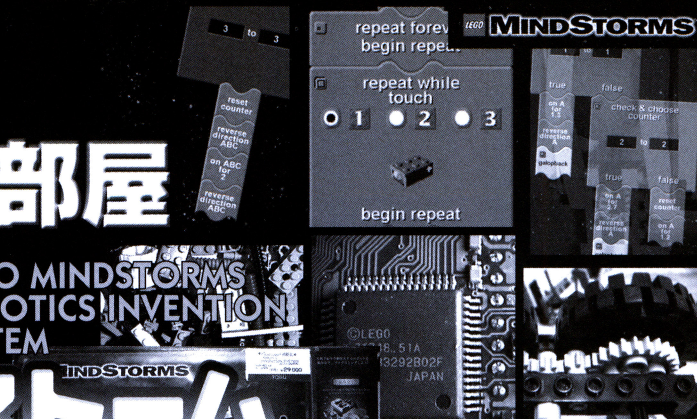
▲お腹のコネクタに専用ケーブルを差してフラッシュROMを更新！

パソコンでロボットを操る！

レゴ道場 桐林部屋

チャレンジ!! レゴ・マインドストーム

LEGO MINDSTORMS ROBOTICS INVENTION SYSTEM



お題：ライン・トレーサ

今回は前号で紹介したセンサの続きにもあたる「ライン・トレーサ」です。RISの説明書にも載っているくらいですので,初心者にも簡単にできるものですが,とても奥が深いものです。（文/桐林星河）



『LEGO MINDSTORMS Robotics Invention System 2.0（日本語版）』
実売価格：約23,800円
対応OS：Windows98/Me
総パーツ数：718（レゴブロック/RCX/USB IRタワー/モータ2個/タッチセンサ2個/ライトセンサ1個/日本語マニュアル/日本語ソフトウエアなど）

ライン・トレーサの種類は大きくわけて2種類

ライン・トレーサには，大きくわけて2種類あります。ひとつは，左右のタイヤの回転数を変えて曲がる方法（以後，駆動分散型）で，もうひとつは，車のようにハンドルを切って曲がる方法です。駆動分散型はその場で回転することができます。そのおかげで小回りが利きますが，長い直線では蛇行してしまいタイムロスになります。

それに対してハンドル型は，カーブがきついと曲がれませんが，直進性がよく直線のスピードがとても速くなります。

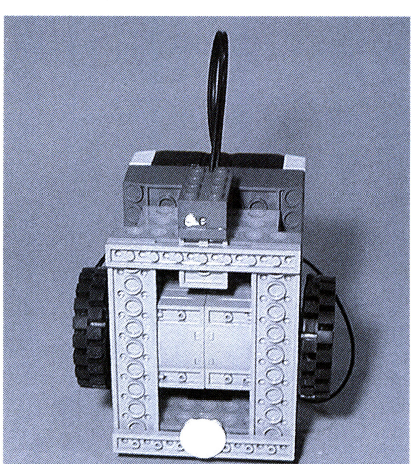
簡単に作れる基本型「駆動分散型」

駆動分散型は，なんといっても仕組みがとても簡単で，時間がかからず簡単に作れるのが特徴です。そのためRISの説明書にも例として登場してきます。一般的なのが左右に1個ずつタイヤとモータをつけ，1個のライトセンサで地面の色を判別して動きます(**写真1**)。動きかたを簡単に考えると**第1図**のようになります。

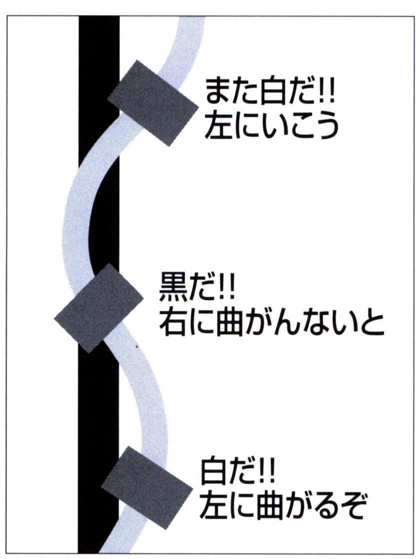
直進に苦手な部分がありますが，プログラムとコースしだいでは，高速走行が可能になります。ライトセンサが二つや三つあると，さらに走行がスムーズにできます（**第2図**）。

第6回お気軽コンテストに参加した「カラトンVer2」は，ライトセンサを三つ付け，プログラムを工夫して高速走行を可能にしました。RIS1.5のテストパッドを3周するのにわずか15秒ほどしかかかりません。これは駆動分散型では最速だと信じています（^^ゞ。

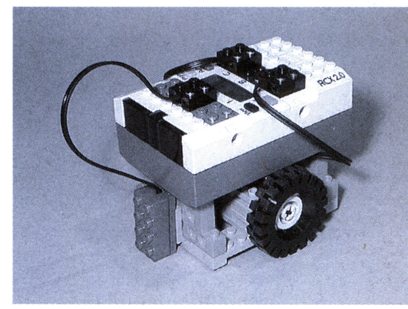
今回の実験用のマシンは簡単にするため，RISに入ってくる大きめのタイヤをモータに直接つけただけの単純なものです（**写真2**）。



《写真1》一番上の突起がライトセンサ



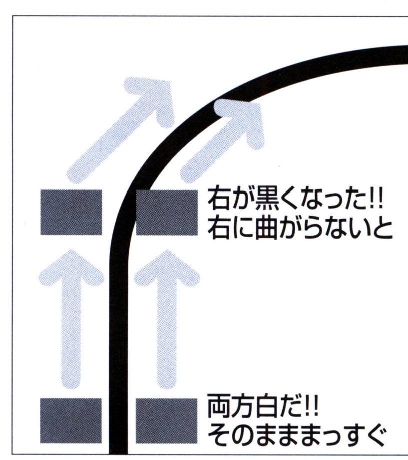
《第1図》駆動分散型の動き。四角いものがライトセンサ



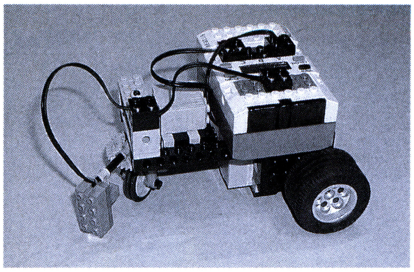
《写真2》今回の実験用の簡単なライン・トレーサがこれ

構造は複雑で奥深い「ステアリング型」

ステアリング型はRISのセットだけではなかなか作るのがむずかしいのです



《第2図》センサを増やすとスムーズになる



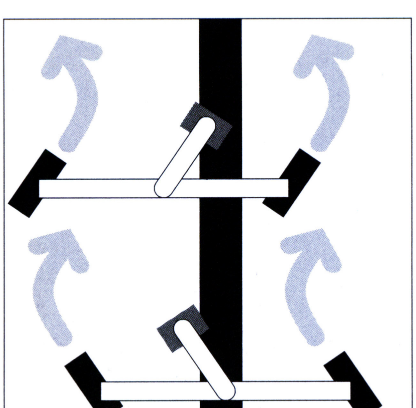
《写真3》ステアリング型のライン・トレーサ。RISのセットだけで作れる

が，センサ1個でも，とても速いマシンが作れます。

しかし，構造が少し複雑になるので，駆動分散型よりも大きくなりがちです(**写真3**)。ステアリング型の動きかたは**第3図**に示します。やはりステアリング型のマシンであっても，センサが多いほうがスムーズに動きます。これは駆動分散型と同じ原理です。しかし，ステアリング型の場合は，処理速度がある程度速くないと，すぐにコースアウトしてしまうという弱点があります。

ステアリング型の場合，もう一つの問題が「内輪差」です。内輪差は車がカーブを曲がるとき，外側のタイヤと内側のタイヤで進む距離が変わってしまうことです(**第4図**)。そこで，たいていはデフギヤを入れて内輪差を吸収します。しかし，カーブが右回りか，左回りのどちらか一方しかない場合は，片側のタイヤだけ駆動して，反対側を空転するようにしておけば，内輪差は左右のタイヤの軸がつながっていないので起きません。

もうひとつのポイントが，ハンドル用のモータです。ハンドルは，一定の角度以上は曲がることがないので，タッチセンサなどでハンドルが端に来たことを感知したり，クラッチギヤを使って，無理な力（ハンドルを端まで切ったとき）がかかったときに空回りさせないと，モータに負荷がかかってしまいます。



《第3図》ステアリング型。真ん中の四角いのがライトセンサで，左右の四角いのが前輪。駆動分散型のものとほとんど同じ

今回は部品の制限などもあったので，両方ともつけませんでした。そのため，コースアウトしてしまうと，ステアリングモータに負荷がかかってしまいます。もしコースアウトしたらすぐに止めてください。

プログラムの処理速度が総合的に影響

ここまで紹介したように，マシンの仕組みだけでかなりスピードが変わります。しかしライン・トレーサの場合，一番影響するのがプログラムの「処理速度」です。処理速度はパソコンでも同じですが速ければ速いほど計算がすぐ終わり，結果がすぐに出せます。つまり，ライン・トレーサの場合は「黒い線を見つけた!!」という部分から「曲がらないと!!」という結果までの間が短いことになり，黒い線をライトセンサが通り越すということが少なくなります。

現在よく使われているプログラムの処理速度を速い順にならべてみます。

1. legOS
2. NQC, LOBOLAB, RIS2.0
3. RIS1.5, RIS1.0

見てのとおり，RIS標準のRCXCODEは遅い分類に入ります。しかし，RIS2.0では処理速度が格段に上昇しています。1番のlegOSは，自分でファームウエアごと作ってしまう上級者向けのプログラムです。なので，ここではあえて紹介しません。

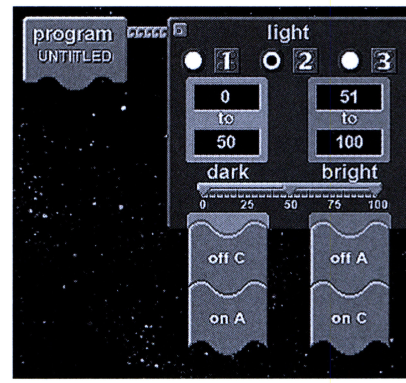


《第4図》四角いものがタイヤ。外側と内側では回る距離がぜんぜんちがう

僕は普段NQCとRIS1.5のRCXCODEを使いわけています。簡単なものはRCXCODEで，すこし処理が難しくなるものはNQCでプログラムしています。

プログラムによるスピードのちがいを実験

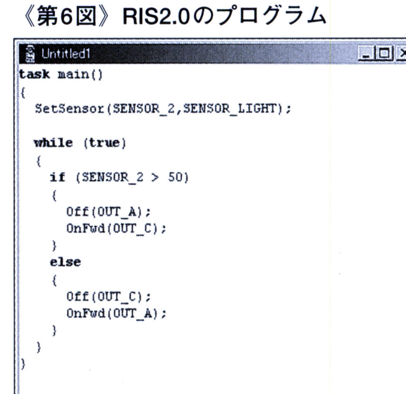
では実際に簡単な駆動分散型のライン・トレーサと，ステアリング型のライ



《第5図》RIS1.5のプログラム



《第6図》RIS2.0のプログラム



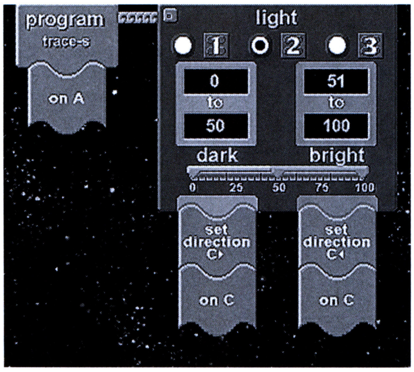

```
task main()
{
  SetSensor(SENSOR_2,SENSOR_LIGHT);

  while (true)
  {
    if (SENSOR_2 > 50)
    {
      Off(OUT_A);
      OnFwd(OUT_C);
    }
    else
    {
      Off(OUT_C);
      OnFwd(OUT_A);
    }
  }
}
```

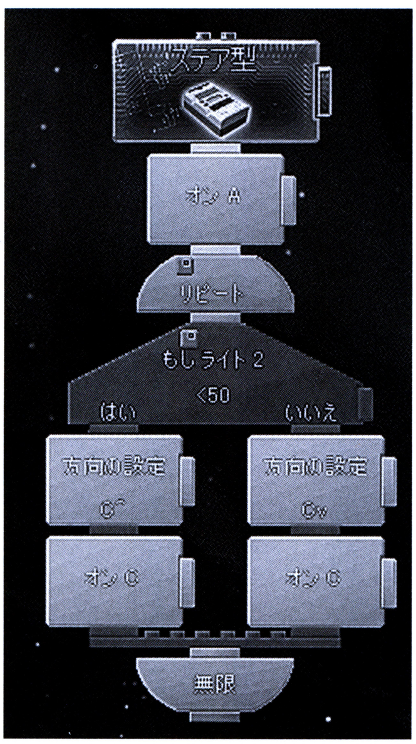
《第7図》NQCでのプログラム

ン・トレーサを作って，プログラム言語によるスピードのちがいなどを見てみましょう。

駆動分散型で三つの言語で試してみました。**第5図**はRIS1.5のプログラム，**第6図**はRIS2.0のプログラム，**第7図**はNQCのプログラムです。処理の内容はすべて同じで，「黒い線の上に来たときに右に曲がれ，白くなったら左に曲がれ」というものです。結果は**第1表**を見てください。



《第8図》ステアリング型のRIS1.5のプログラム



《第9図》ステアリング型のRIS2.0のプログラム

```
task main()
{
  SetSensor(SENSOR_2,SENSOR_LIGHT);
  OnFwd(OUT_A);
  while (true)
  {
    if (SENSOR_2 < 50)
    {
      OnFwd(OUT_C);
    }
    else
    {
      OnRev(OUT_C);
    }
  }
}
```

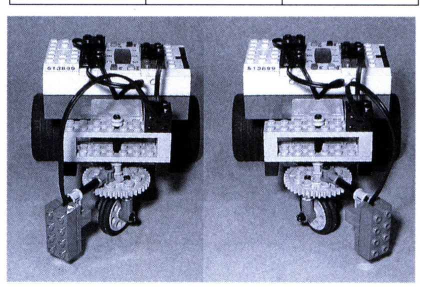
《第10図》ステアリング型のNQCのプログラム

つぎにステアリング型です。ステアリング型のプログラムもRIS1.5（**第8図**），RIS2.0（**第9図**），NQC（**第10図**）で作ってみました。処理の内容は「黒い線の上にきたらハンドルを右に，白いところにきたらハンドルを左に切る」というもので，駆動分散型とほぼ同じです。マシンはいちばん簡単に作れる3輪車型にしました。前輪を1輪にして，水平なギヤに直接タイヤをつけ，ギヤを回転させることによりハンドルを切る仕組みです。このハンドルには直接ライトセンサがついています（**写真4**）。

どうですか？　プログラムによってずいぶん変わるのがわかりましたね?

《第1表》RIS1.5に付いてくるテストパッドを3周回ったときのタイム表

プログラム	駆動分散型	ステアリング型
RIS1.5	低速にして走行可	走行不可
RIS2.0	22.20秒	14.98秒
NQC	21.95秒	14.52秒



《写真4》ハンドルを切っているところ。タイヤとセンサは直結している

さて次号は

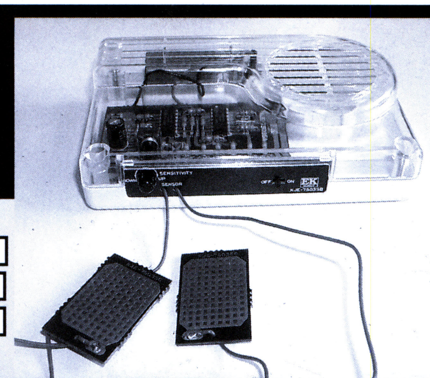
次回は7月22日に行われた「ツクモMUGカップ ライントレースロボット競技会」に参加したマシンの紹介です。

この競技会はライン・トレーサの競技会なので，今回はライン・トレーサの解説にしてみました。基本的な仕組みを知ってから，別のマシンを見ると，さらにおもしろさが伝わると思います。お楽しみに!!!

松原拓也の自作一代記（じさくいちだいき）

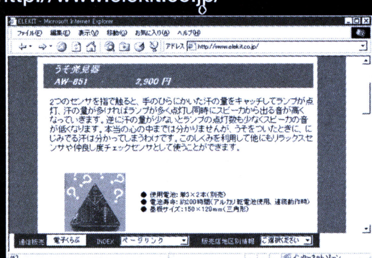
連載6回目 エレキットで遊ぼう

「うそ発見器」の製作



「電子工作はむずかしそう」と思っている人には電子工作キットがおすすめです。電子工作キットは材料がすべて用意されているセットのことで，道具を用意すれば簡単に組み立てることができ，電器回路の入門に最適です。そういうわけで，今月は電子工作キットを使った自作を紹介します。 （松原拓也）

イーケイジャパン「ELEKIT」WEBサイト
http://www.elekit.co.jp/



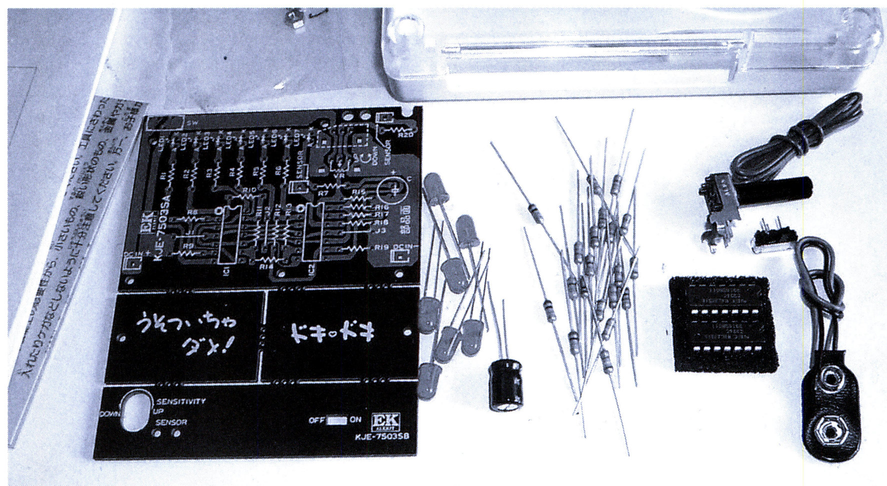
▲ELEKIT「うそ発見器（AW-851）」
価格2,900円（税別）



▲ELEKIT「うそ発見器Ⅱ（JE-7553）」
価格2,600円（税抜）

電子工作キットの老舗といえば，イーケイジャパンの「エレキット（ELEKIT）」が有名です。「エレキット」はロボットやゲーム機，そしてセンサ類など膨大な量の製品が揃っています。そうしたラインナップの中で，古くから異彩を放ってきたのが「うそ発見器」です。「うそ発見器」はその名のとおり，人のウソを見破る装置で，電子ブロックの工作例でもおなじみです。

エレキットには「うそ発見器（AW-851）」と，「うそ発見器Ⅱ（JE-7553）」という2種類の「うそ発見器」があります。両者の精度は同じですが，「うそ発見器Ⅱ」は「うそ発見器」に比べ，定価が300円ほど安いかわりに，音が鳴る機能が省かれています。



▲《写真1》「うそ発見器Ⅱ」に入っている部品

どちらがいいか悩みましたが，結局「うそ発見器Ⅱ」を東急ハンズで購入しました。

道具の用意

製作に入る前に，まず以下の道具を用意します。

・はんだ
・はんだこて&はんだこて台
・ニッパ
・カッターナイフ

それと，キットには電池がついてきませんので，006P（9V）電池を1個用意しておきましょう。

キット内容

このキットには以下の部品と説明書が入っています（**写真1**）。

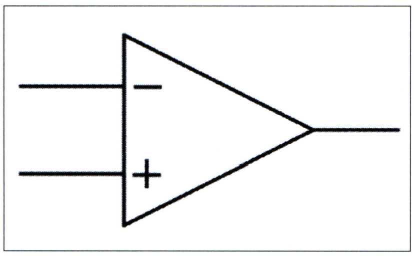
・抵抗
・LED（発光ダイオード）
・電解コンデンサ
・スイッチ
・IC（オペアンプ）
・基板
・電源タップ
・リード線

回路の説明

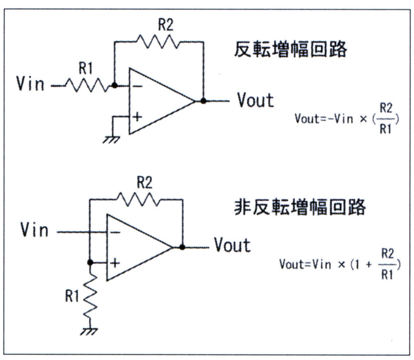
人間の手と手の間には，2k～10kΩ程度の抵抗値があり，そこに電気を流すことができるそうです。キットの「うそ発見器」はこの抵抗値をLEDの点灯として表すものです。もし電圧が一定だと，抵抗値の変化は電流の変化になります（オームの法則）。この抵抗値を電位に変えて増幅します。増幅はオペアンプ（**第1図**）という部品で行います。オペアンプの非反転増幅回路（**第2図**）で増幅された電位はLEDに7段階の点灯として出力されます。この電圧の比較でもオペア

ンプが活躍します。オペアンプは「＋」「－」の入力を比較して，その結果を出力する機能があるためです。

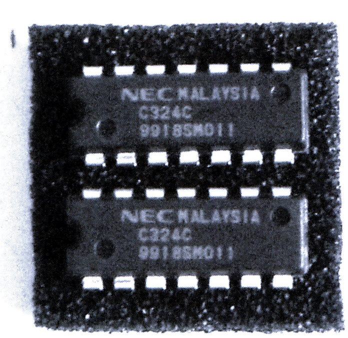
使用しているオペアンプはNEC製の「uPD324」です（**写真2，第3図**）。動作電圧は3～30Vなので，3端子レギュレータなしに，電池の9VをそのままVCCに使っています。



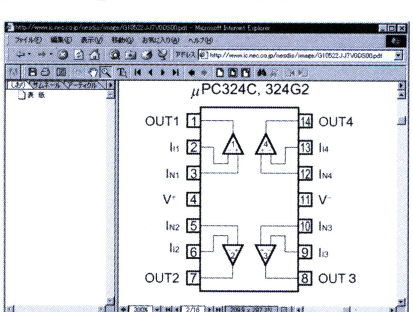
《**第1図**》オペアンプの回路図記号



《**第2図**》オペアンプを使った「反転増幅回路」と「非反転増幅回路」



▲《**写真2**》NEC製のオペアンプ「uPD324」。リニアICとも呼ぶ

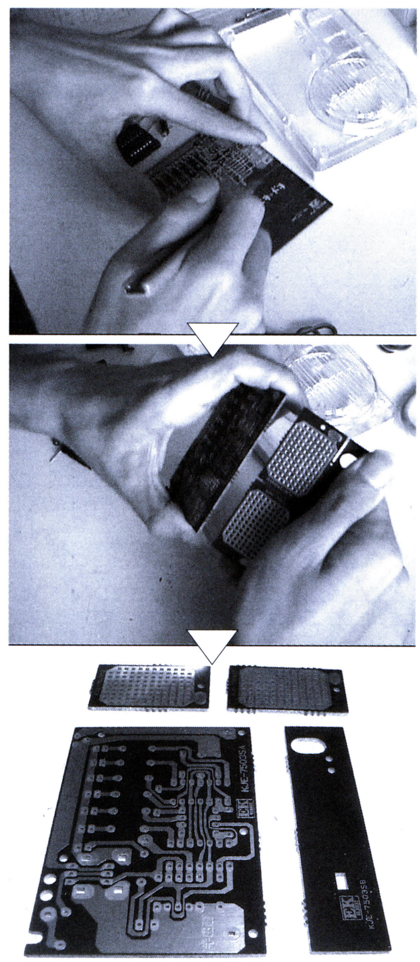


《**第3図**》「uPD324」の仕様書。DIP14pinパッケージにオペアンプが4個内蔵されている

組み立て

まず，基板にカッターナイフで傷をつけ，折って4枚に分離させます（**写真3**）。

そして，説明書に記述しているとおりに基板に部品を取り付けます。はんだ付けの順番ですが，背の低い部品から付けていくといいでしょう（**写真4，5**）。ICとコンデンサとダイオードには極性があります。基板に印刷されているシルク印刷や説明書を参考にして，向きをまちがえないように取り付けましょう（**写真6，7**）。

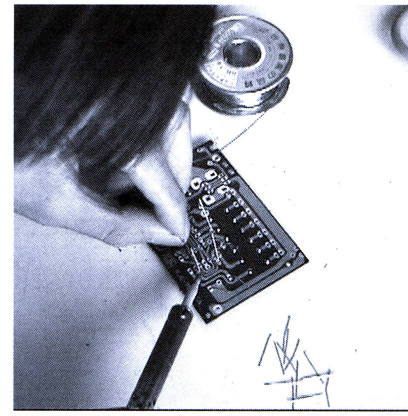


▲《**写真3**》基板を折って分離する

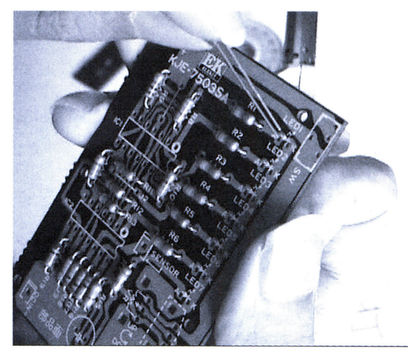


▲《**写真4**》説明書を見ながら部品を取り付ける

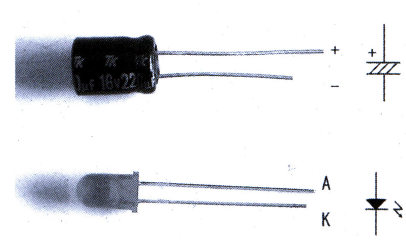
あとは，可変抵抗，ジャンパ，スイッチを取り付けて（**写真8**），基板とリード線を加工してセンサを2個取り付けたら完成です（**写真9**）。



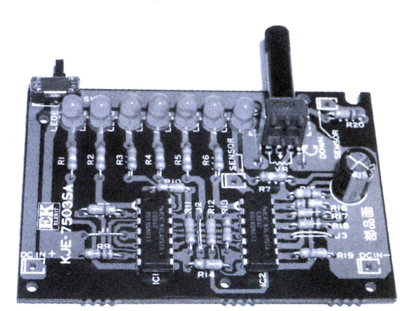
▲《**写真5**》背の低い部品から，はんだ付けするといい



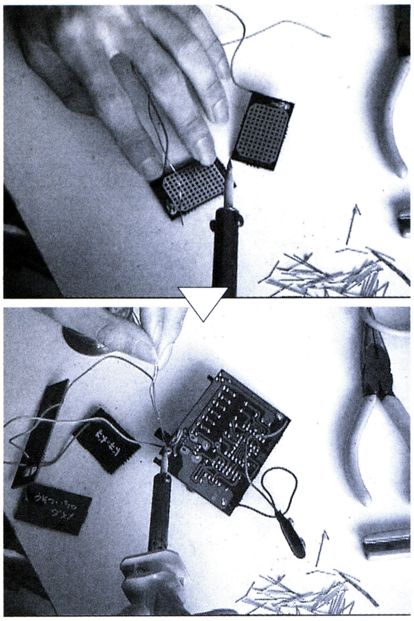
▲《**写真6**》LEDやコンデンサの取り付けは極性に注意する



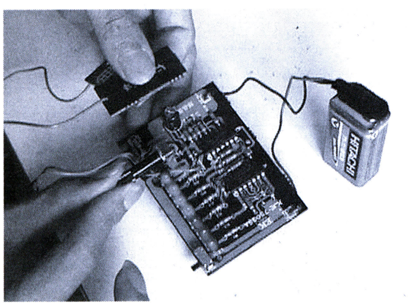
▲《**写真7**》電解コンデンサとLEDの極性



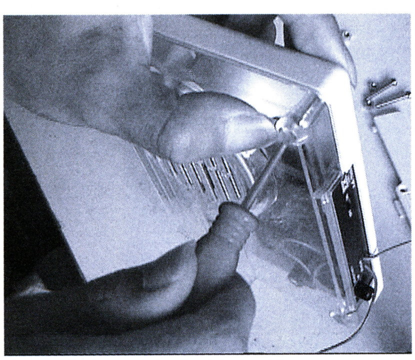
▲《**写真8**》可変抵抗，スイッチ，ジャンパの取り付けが完了した状態



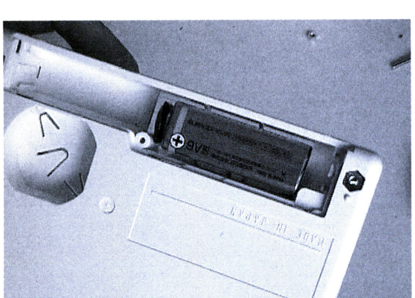
▲《写真9》センサの取り付け



▲《写真10》動作テストのようす。可変抵抗に応じてLEDが付けば成功



▲《写真11》基板を外装に組み込む



▲《写真12》電池の穴を広げて9V電池が入るように加工する

動作テスト

9V電池を装てんして電源を入れます。ためしにセンサの両端を合わせて，LEDを点灯すれば成功です。感度の調整は可変抵抗のボリュームを回転させます。

2個のセンサ間の電流が増すと，LEDの点灯する数が増えます。逆に流れが悪くなると点灯する数が減っていきます。LEDは七つの段階で表示されます（**写真10**）。

外装に組み込み

基板をケースに組み込みます（**写真11**）。9V電池の装てんする場所は，プラスチックのパーツを削りとります（**写真12**）。

ついに「うそ発見器Ⅱ」の完成です（**写真13**）！

うそ発見器を使う

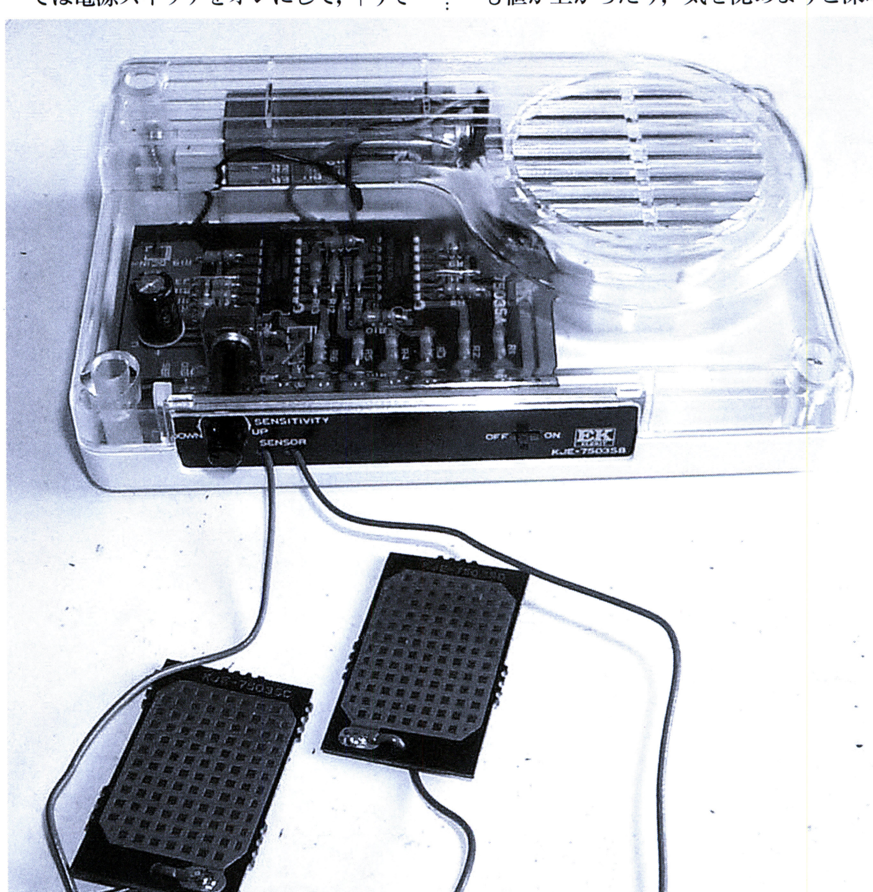
では電源スイッチをオンにして，「うそ発見器」を使ってみましょう。「うそ発見器」のセンサは片手で持つか，両手に片方ずつ持つかどちらでも構いません（**写真14，15**）。そのつど最初にボリュームを回して，点灯するLEDの数を少なめに調整します。

そして質問に答えます（ここでうそをつく）。さすがに，自分で自分に問題を出せませんので，別の人にお願いして質問をしてもらいました。質問内容は，たとえばこんな感じです。

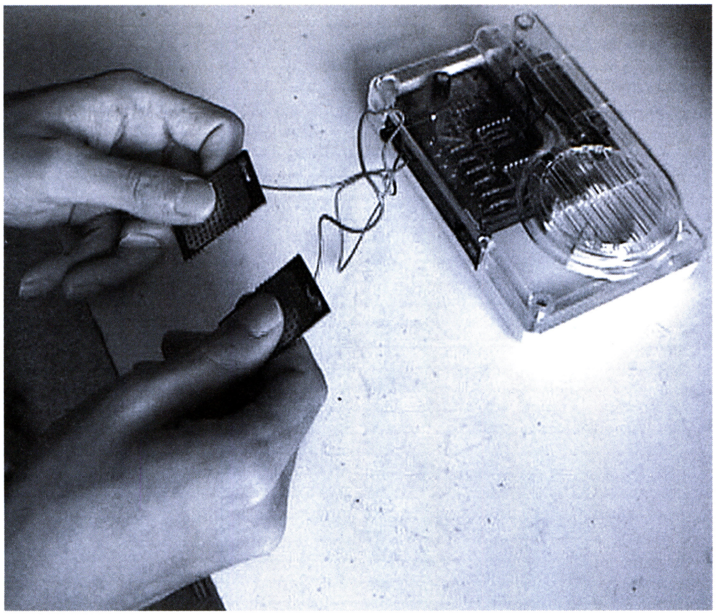
質問「昼にラーメンを食べましたか？」
被験者「はい」（これはうそ）

質問「今日は昼過ぎまで寝てましたか？」
被験者「いいえ」（これもうそ）

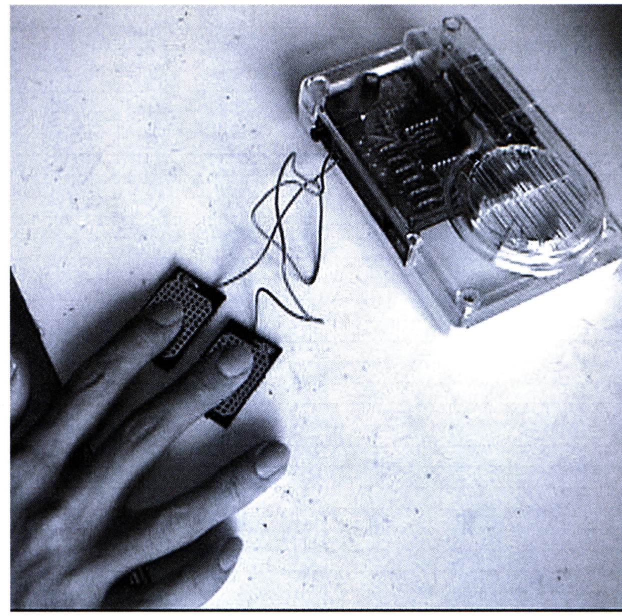
……と，こうした質問に答えて3～4秒ほど待つと，点灯しているLEDの数が一つ増えました。変化は微妙なもので，「なんとなく」うそを判定している感じがします（本当になんとなくです）。しかし，センサを握る手の湿気で，放っておいても値が上がったり，気を沈めようと深呼



▲《写真13》完成した「うそ発見器Ⅱ」



▲《写真14》測定中。センサを両手に持った場合



▲《写真15》センサを片手で持った場合

吸したら，逆に値が上がってしまいました。また，センサの指をちょっとでも動かすと，値に影響してしまいます。

このように測定結果が非常に不安定で信憑性がありません。結果はあくまで「遊び」としてとどめておいたほうがよさそうです。

動作原理

この「うそ発見器」は，

①うそをつく
↓
②緊張して汗をかく
↓
③電気が流れやすくなる
↓
④LEDが点灯する

……というメカニズムで動作します。

しかしなぜ「うそ」をつくと汗をかくのでしょうか？　人間の体には自律神経という器官があり，体温調節や発汗，呼吸などをつかさどっています。自律神経は人体を活動的にする交感神経と，逆に沈静させる副交感神経の2種類にわかれています。そして「うそ」をつくと，緊張や動揺が起きて交感神経が興奮します。その刺激は汗腺に伝わって，手のひらや腋の下などで汗が分泌されます（ゲームに熱中して手に汗をにぎるのと同じ)。

こうした変化の電気による反応を皮膚電気反射（GSR：Galvanic Skin Reflex）といいます。

余談ですが，警察が科学捜査で使っている「うそ発見器（ポリグラフ)」は皮膚電気反射と一緒に，呼吸，血圧，脈拍を組み合わせて測定します。しかし測定結果が微妙で，結果を自白の強要に使われたりと問題点が多いため，証拠能力はほとんどありません（それでも最高裁で証拠と認められた例がありますが)。

さらに余談ですが，イスラエルのTrustechというメーカーが，音声でうそを分析する「Truster（トラスター)」という製品を販売しています（パソコンソフト版と携帯機版の2種類があります)。「的中率85%以上」と広告で豪語していますが，果たしてその的中率は本当なんでしょうか？

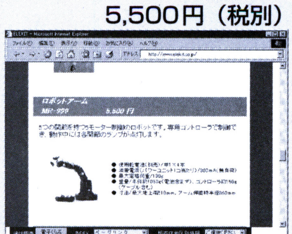
まとめ

エレキットは入手が簡単で，説明が親切丁寧なので，電気工作の入門用として最適かもしれません。

制作費2,730円（税込)。製作時間は約1時間30分でした。

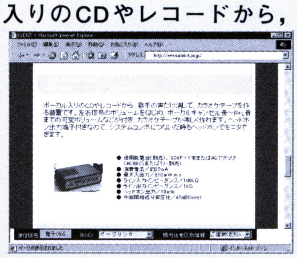
■その他の「エレキット」　膨大なエレキットの製品群の中から個人的に気になったものを選んでみました。

●「ロボットアーム（MR-999)」
5,500円（税別）



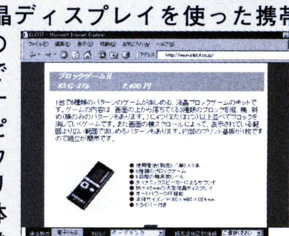
五つの関節を持つロボットアームです。内蔵されたモータは五つ。専用コントローラでリモコン操作します。別売りの「ムービット・ラボ（IF-99)」があれば，パソコンからのプログラミング（BASICに似たオリジナル言語）で制御することができます。「ムービット・ラボ」の対応OSはWindows95で16,000円（税別）

●「カラオケ新工場（EL-556)」
4,500円（税別）



ボーカル入りのCDやレコードから，歌手の声だけ消してしまう装置です。カラオケテープを作るのに便利。ヘッドホン出力端子付きです。

●「ブロックゲーム（KEG-275)」
2,400円（税別）



大型液晶ディスプレイを使った携帯ゲーム機のキットです。ダイナミックスピーカでサウンドが鳴ります。本体サイズはH180×W80×D24mm。ゲームは，落ちモノ系のパズルなど6種類です。

テクノ・トイズ・ラボラトリ
Techno Toys Laboratory

デジタルとアナログの 境界線上で作る電子オモチャ

今月の製作 デューパールーパ 試作編

テクノ・オモチャ研究所

TEXT&製作●クワクボリョウタ



デバイス・アーチスト。1998年頃から「ビット」をキーワードにメディアアート作品や電子オモチャなどを製作・発表。最近いろんなアイディアが溢れ出てきてコマッチャウ。時間と資金と体力だけが足りない今日この頃。www.vector-scan.com

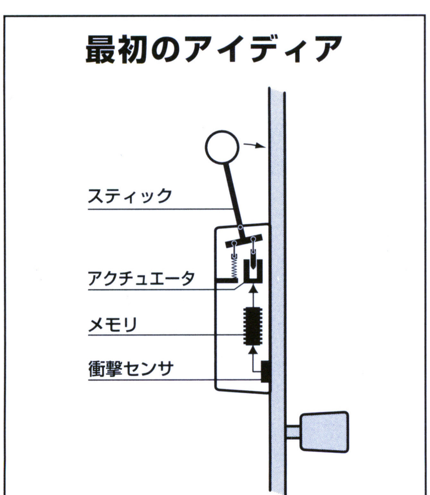
当研究所はいよいよ三つ目のアイテムの製作に入る。今回はドラムマシンの研究だ。とは言ってももちろん，ただのドラムマシンではなくて，実際にモノを叩く機械式のリズムシーケンサを開発してみた。今回はプロトタイプ編です。

ヒントはトイレから？

トイレのドアをノックした経験はだれにでもあると思うけど，あれってなんかデジタル通信っぽくない？　半2重シリアル通信そのものじゃないかと思うんだけど。

まあ，それはおいといて，今回の製作のヒントはまさにトイレのノックから得たものなんだ。公衆トイレで順番待ちしながら，「ドアの内側に，ノックの音を検出して同じ音のパターンを真似て返事するような装置をつけたら，だれも入れなくて面白いなぁ」なんてことを考えついたのだが，製作のネタというものはこんなふうに日常に転がっているわけです。

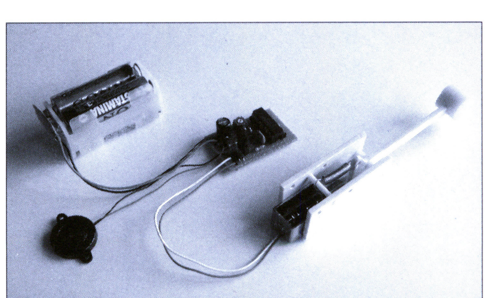
で，早速その装置のブロック図を書いてみると**第1図**のようになる。仕組みはシンプルで，ノック音を検出するセンサとノックをするためのアクチュエータ，それから検出した音のパターンを遅らせて返事らしいタイミングをとるためのメモリで構成される。書いてみて気がついたことは，装置のノックの出力が自分の入力にフィードバックされた場合，こいつは同じパターンを延々と叩き続けるのではないかということだ。これってドラムマシンになるかも？　と気づいのだが，えてして製作はこんなふうに思わぬ方向にズレていくのだった。



《第1図》最初に思い浮かんだ装置いつまでもノックを返すトイレの構想…。今回製作する装置も仕組みはまったく同じだ

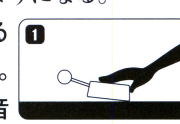
デューパールーパの使いかた

こうして構想に至った反復型ドラムマシンはデューパールーパと名付けられた。打音を複製（デュープ）して反復（ループ）することに由来している。システム構成はトイレのイタズラ装置と全く一緒で，衝撃センサからの入力をディレイしてソレノイド（電磁石により往復運動をするアクチュエータ）から出力する。**写真1**が製作したデューパールーパのプロトタイプだ。電源，回路，センサ，ソレノイドによるドラム・スティックのメカから成り立っている。

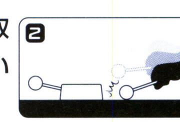


《写真1》センサとスティックの位置関係によって動作の具合も変わってくる

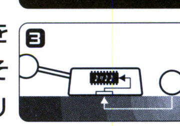
使いかたは以下のようになる。



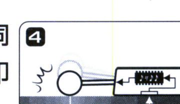
①本機を何か叩かれる部材に取り付ける。ドア，床，箱など音がするものなら何でも可。



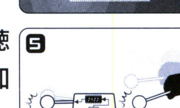
②プレーヤが本機を取り付けた部材を叩いて音をだす。



③叩いた衝撃は部材を伝わり，センサがそれを検出してメモリに蓄える。



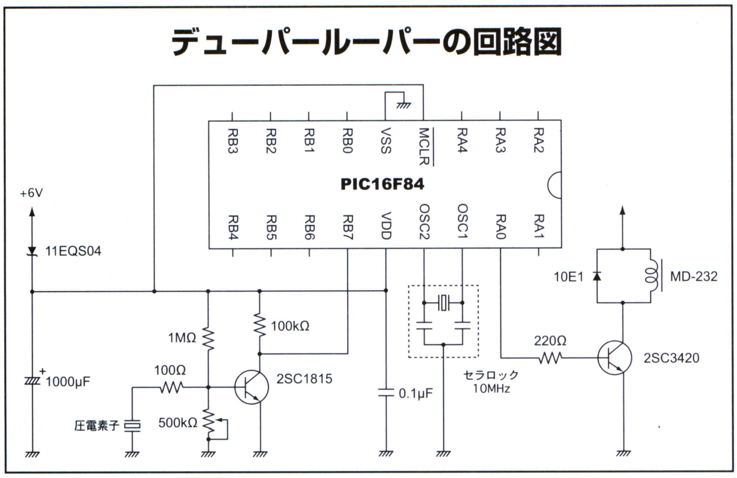
④1小節後，本機が同じリズムで部材を叩きだす。



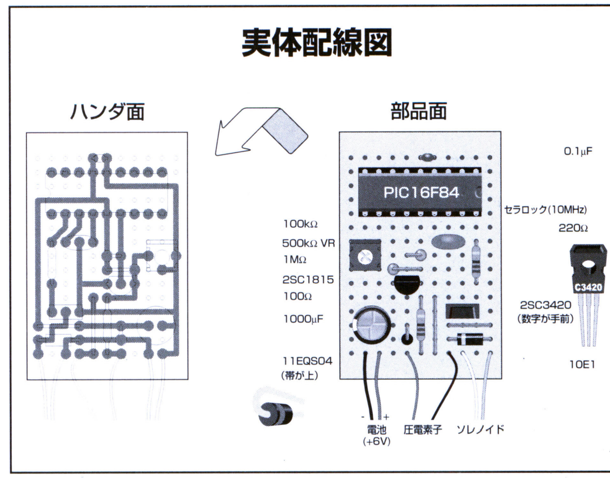
⑤プレーヤはそれを聴きながら打音を追加することができる。



⑥スティックを手でおさえることで記憶された打音を消去できる。



《第2図》図中の↑はともに電池に接続する



《第3図》取り付け方向がある部品にはとくに注意してハンダ付けしよう

文章で書くと複雑めいてしまうけど，実際にはとても簡単かつ直感的な操作で演奏できる。プレーヤは部材を叩くか，デューパールーパのスティックを手でおさえるだけ。ちなみにこのような，ディレイのかかった音にさらに音を重ねる演奏方法はサウンド・オン・サウンドと呼ばれる。

回路図

デューパールーパの試作回路図は**第2図**，実体配線図は**第3図**を見てもらいたい。そして，パーツリスト**第1表**を参照のこと。これは試作版なので，次回完成編で若干の部品の追加をするけど，大きな変更はないので安心して組み立てを開始していただきたい。

センサは以前も使った圧電素子をつかう。圧電素子は電圧を加えると音が出るのでブザーとしても使えるが，逆に振動を与えると電圧を発生する（**第4図**）。今回は後者のようにセンサとして使う。打音を音として検出すると色んな音に反応してしまうので，叩く部材の振動を直接拾うようにする。

トランジスタで圧電素子の微弱な出力電圧を増幅し，PIC（今回もPIC16F84を使う）の入力ピンに入力する。その際，感度調整を半固定抵抗で行う。

増幅回路の関係で信号の高低は，普段はHi（高い電圧）で衝撃を検出するとLo（0V）になる。

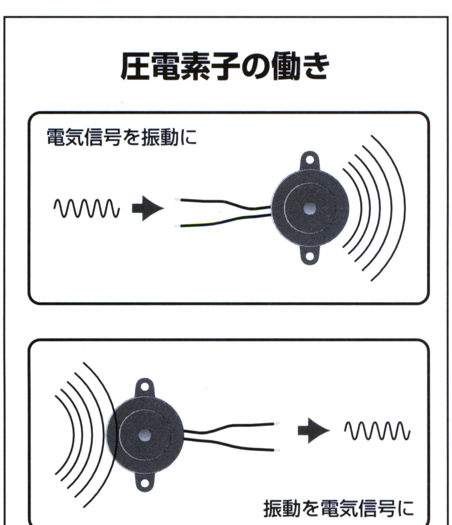
部材を叩いて音を出すには電気信号を機械運動に変える必要がある。そのために使うのがソレノイドだ。ソレノイドは電磁石と鉄芯からなり，電流を流すと鉄芯が吸い寄せられて往復運動を作り出す（**第5図**）。スティックでものを叩かせるには，後で説明するように，この往復運動を機構を作ってスティックの回転運動に変換すればよい。

《第1表》デューパールーパのパーツリスト

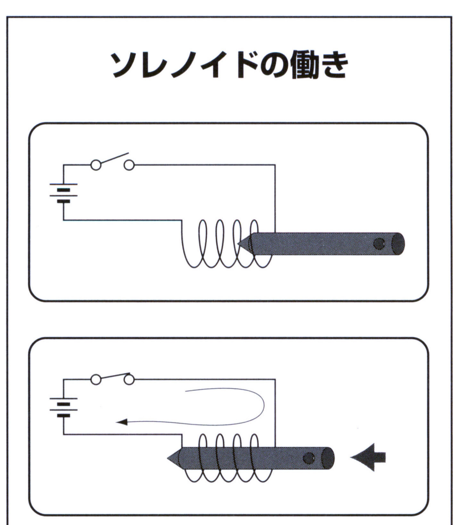
ユニバーサル基板（45mm×45mm程度のものを切って使用）		×1
電池ボックス（単3乾電池4本用）		×1
18ピンICソケット		×1
PIC16F84		×1
セラロック 10MHz（無い場合はクリスタル10MHzとセラミックコンデンサ20pF x 2でも可）		×1
トランジスタ	2SC1815	×1
	2SC3420	×1
ダイオード	10E1	×1
ショットキ・バリア・ダイオード	11EQS04	×1
抵抗	100Ω	×1
	220Ω	×1
	100kΩ	×1
	1MΩ	×1
半固定抵抗	500kΩ	×1
積層セラミックコンデンサ	0.1μF	×1
電解コンデンサ	1000μF 16V	×1
圧電素子		×1
ソレノイド MD-232-1（6V用）		×1
その他、プラ版やバネなど		

ソレノイドが動くためにはとても大きな電流を流す必要がある。PICでは直接ドライブできないので，ここにもトランジスタを使う。ソレノイドの両端につけられたダイオードは電磁石に発生する逆起電力からトランジスタを守るためのものだ。

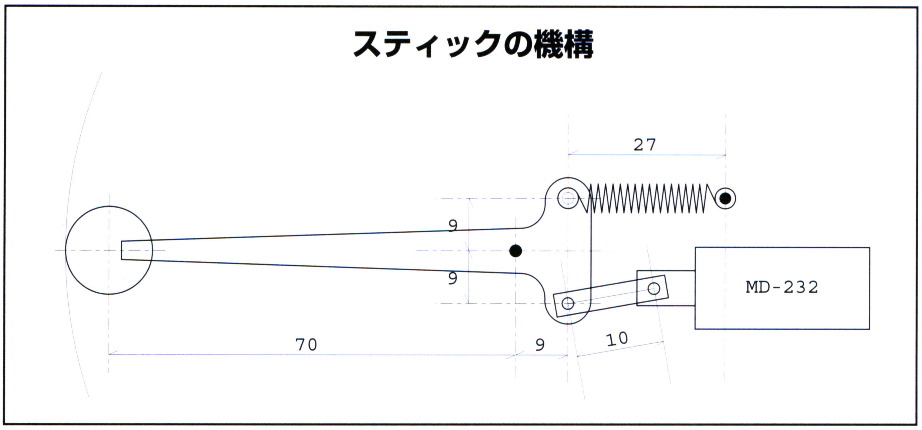
ソレノイドはとても大きな電流を消費するので，電源の扱いに注意が必要になる。電磁石がONになると，電池の容量の関係で電源電圧が落ちてしまう。これはPICの誤動作につながるので，ダイオードとコンデンサでそれを防ぐ。そして，電池はアルカリ乾電池を使ったほうが無難だ。



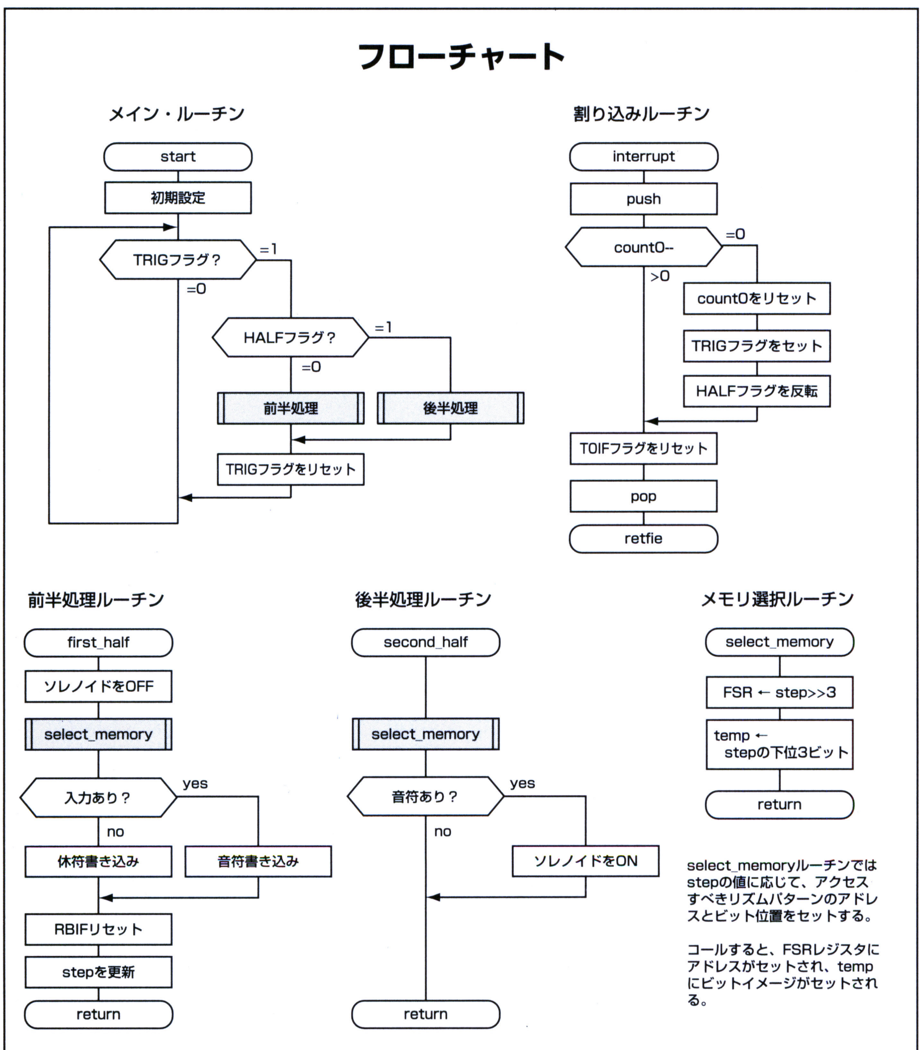
《第4図》圧電素子は電気信号を加えると音を発生する。逆に，振動を与えると，電圧を発生する



《第5図》ソレノイドは電流を運動に変える。電流を流すと鉄芯が吸い寄せられ，往復運動が生まれる



《第6図》ソレノイドと黒丸の部分はシャーシに固定する。適切な寸法は入手したバネの強さによっても変わってくるので注意しよう



《第7図》フローチャートをみてもわかるとおり，プログラム自体はいたってシンプルだ

スティック部の組み立て

ソレノイドの往復運動をスティックの回転運動に変える機構はまさに工作の世界。ここはプラ板とプラ棒を切って製作した。この部分の完成形の図面は次回に掲載したいと思う。というのも，この部分の設計次第で音のシャープさが決まってくるようなので，実はいろいろと試行錯誤中なんですな。参考までにおおよその作り方を**第6図**に示すけど，ここは各自工夫してみてほしいところ。機構の寸法，バネの強さやスティックの作りにいたるまで，ひとつひとつのちょっとした違いが動作に影響してくる。意外と奥が深いところなのだ。

とはいってもセンサが反応するだけの音を出すのはそれほど難しいことではないので，ありあわせの材料でまずはやってみて。

プログラムの概要

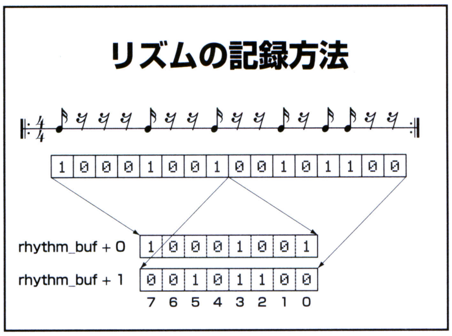
第7図がテストプログラムのフローチャート，**リスト1**がそのソースリストだ。PICに書き込むときは，HSモード，パワー・アップ・タイマ使用，ウォッチドッグ不使用と指定してください。プログラム的にはたいしたことはやっていないけど，とくに気を使った点を説明しておこう。

ドラムマシンである以上，時間の管理がトテモ大事なのはいうまでもない。タイマ割り込みによって一定時間を測り，フラグを通じてメインルーチンへ処理のタイミングを知らせる作りになっている。

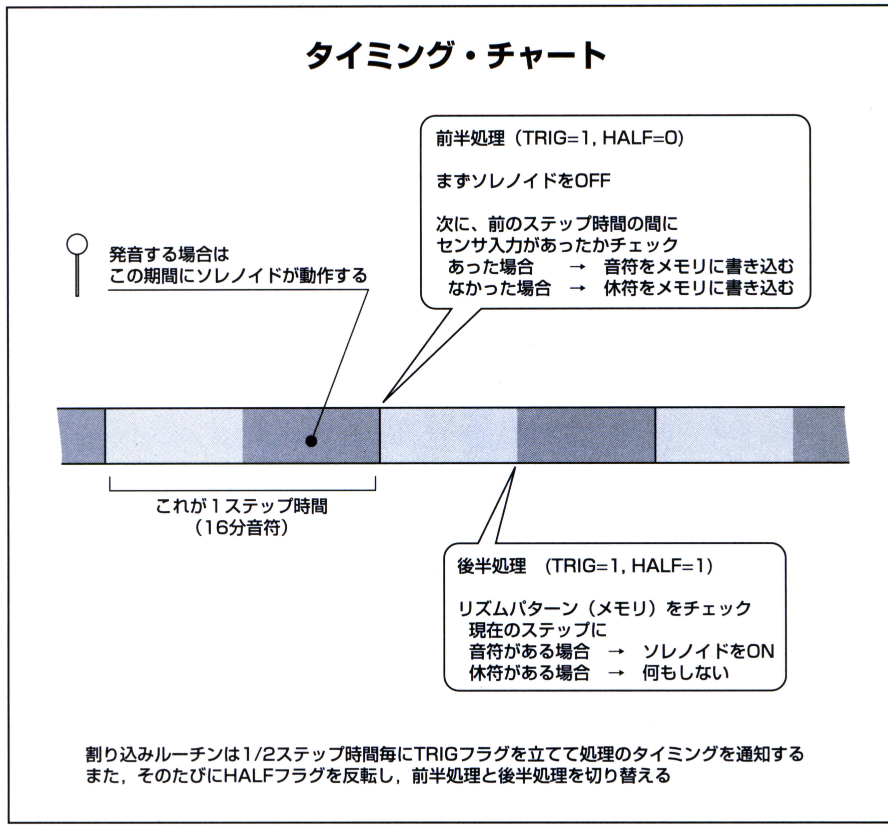
記憶するリズムパターンは16ステップで一回りするように設定した。リズムパターンのメモリ管理は**第8図**のようになっている。

一定時間毎に参照するステップをひとつずつずらしながら，音符があるときはソレノイドを動かし，休符のときは動かさない。一方で，各ステップ毎に衝撃センサからの入力があったかどうかを調べ，入力があった場合は現在の参照ステップに音符を書き込み，入力がなかった場合は休符を書き込む。これを繰り返す。本機がソレノイドの運動によって部材を叩くと，その衝撃が自分自身のセンサに入力されるから，結果的に同一パターンがループしていくことになる。また，ソレノイドの動きを強制的に手で止めると，センサ入力はなくなるから，休符が書き込まれる。このことにより，ループするリズムパターンから任意の音符を取り除くことができる。

そのプロセスを詳しく示したタイミング・チャートが**第9図**だ。各ステップ時間の最後にセンサ入力があったかどうか



《第8図》16ステップのリズム・パターンを2バイトに収納している



《第9図》一定時間毎に2種類（リズムパターンの更新とソレノイドの動作）の処理を交互にくり返している

を調べ，リズムパターンのメモリに結果を書き込む。したがって，プレーヤの打音のタイミングが多少ばらついていても，このステップ時間内に収まっていれば，ジャストのタイミングと同じ扱いになる。16ステップのループを4拍子1小節と考えると，どんなリズムも16分音符単位に直されることになる。このようなテクニックは，コンピュータ・ミュージックの世界では「クォンタイズ（量子化）」と呼ばれる。

ソレノイドの動作開始は各ステップ時間の真ん中で行っている。プレーヤは本機の叩くリズムを聞きながら音を重ねていくのだけど，プレーヤの叩くタイミングが前にずれても後ろにずれても同一のステップに収まるようにするためだ。こうしたほうが誤入力が少なくなる。

それから，クォンタイズするのにはもうひとつ理由がある。ソレノイドが電気的にONになってから，スティックが部材を叩く瞬間までには多少の遅れがある。ステップの分解能が高いと，この遅れの影響でリズムパターンがどんどん狂ってしまう。クォンタイズによって，遅れがキャンセルされるので，リズムパターンは一定に保たれる。

そんなわけで，実際には半ステップごとに割り込みルーチンがTRIGフラグをたてて，センサのチェック＋リズムパターンの更新（前半処理と表記した）と，リズムパターンのチェック＋ソレノイドの動作（後半処理と表記した）を交互に行っている。

センサ入力の調べかた

センサ入力の検出方法について説明しておこう。圧電素子の電圧はトランジスタによって増幅され，普段はHi，衝撃入力時にLoになる。この信号はRB7ピンに接続されている。PIC16F84はRB4～RB7のどれかの入力電圧が変化したときにINTCONレジスタのRBIFフラグをセットする。この機能は「ある期間内に少なくとも1回入力が変化した」ことを検出するのに便利だ。使いかたは以下のとおり。

1.PORTBを読む（または書く）
2.INTCONレジスタのRBIFフラグをリセットする
3.必要になったらRBIFフラグを調べる

1,2が検出の初期化で，RBIFフラグのリセットの前に一度PORTBにアクセスしておくことをお忘れなく。このアクセス時の値と入力を比較して変化を検出するからだ。

この機能を使えば，リアルタイムに入力を見張っている必要がない。ちなみにこのフラグは割り込みも起こすことができる。

さて動作は？

試作機が完成したら早速動作をチェックしてみよう。

机などの表面にスティック部とセンサ部をテープで軽く固定して電源をONにする。この時点では，何も起きないのが正解。

では机を小突いてみよう。ワンテンポおいてスティックが同じリズムで動き出すはずだ。動かないときや，余計な入力を拾って暴れだすような場合は基板の半固定抵抗を回して感度を調節しよう。スティックが，等間隔で同じパターンを繰り返すようになったら成功！

やってみると，叩く部材は金属の缶などよりも木の板のように残響音の少ないもののほうがうまくいくようだ。残響があまり長いと，つぎのステップにまで音がコピーされてしまい，しまいには16連打状態になってしまう。

テンポは140BPMに設定してある。リスト中の定数を変更すれば，テンポやステップ数などを変えることもできる。いろいろいじってみて。

次回はスティック機構の完成形と，幾つかの機能の追加，そして仕上げ，応用例などを紹介したいと思ってます。こうご期待！

《リスト1》完成版トータル・ライフ・アドバイザのリスト（アセンブラ）

```
;======================================================
;
;              d u p e r   l o o p e r
;               ( p r o t o t y p e )
;
;           c o d e d  b y  ｸ ﾜ ｸ ﾎ ﾞ ﾘ ｮ ｳ ﾀ
;
;======================================================
;
        LIST   C=120, p=16F84, r=dec, x=off, n=0
        include  <p16f84.inc>

;======== マクロの定義 ==================================
push    macro
        movwf  w_temp
        swapf  STATUS,W
        bcf    STATUS,RP0
        movwf  status_temp
        endm

pop     macro
        swapf  status_temp,W
        movwf  STATUS
        swapf  w_temp,F
        swapf  w_temp,W
        endm

;======== シンボルの定義 ================================
;-------- レジスタ名の定義 --------
temp            equ     0x0c
w_temp          equ     0x0d    ; 割り込み時のﾚｼﾞｽﾀ退避用
status_temp     equ     0x0e    ;          //
count0          equ     0x0f    ; ﾀｲﾐﾝｸﾞ用ｶｳﾝﾀ
event           equ     0x10    ; ｲﾍﾞﾝﾄﾌﾗｸﾞ
stat            equ     0x11    ; ｽﾃｰﾀｽﾌﾗｸﾞ
step            equ     0x12    ; ﾘｽﾞﾑのｽﾃｯﾌﾟｶｳﾝﾀ
rhythm_buf      equ     0x13    ; ﾘｽﾞﾑﾊﾟﾀｰﾝ用ﾒﾓﾘ
                                ;   (STEP_MAX/8 byte分使用)

;-------- 各変数の初期値 --------
COUNT0_INIT     equ     130     ; 140BPMにﾘｽﾞﾑを決める
STEP_MAX        equ     16      ; 16ｽﾃｯﾌﾟで1ﾙｰﾌﾟ

;-------- eventのビットマップ --------
TRIG            equ     0       ; 半ｽﾃｯﾌﾟ時間経過ﾌﾗｸﾞ

;-------- statのビットマップ --------
HALF            equ     0       ; 前半/後半処理の切替ﾌﾗｸﾞ

;-------- PORTAのビットマップ --------
STICK           equ     0       ; ｽﾃｨｯｸ用ｿﾚﾉｲﾄﾞの駆動制御ﾋﾟﾝ

;======== リセットベクタ ================================
        org    0x0000           ; 起動時は0番地から実行
        goto   initialize       ;

;======== 割り込みベクタ ================================
        org    0x0004           ; 割り込みが起きると
                                ;          4番地から実行
;======== 割り込みルーチン ==============================
interrupt:
        push                    ; ｺﾝﾃｷｽﾄの保存

        decfsz count0,F         ; count0--==0なら以下を実行
        goto   int_90           ; それ以外は90へ

        movlw  COUNT0_INIT      ; count0の初期化
        movwf  count0           ;

        bsf    event,TRIG       ; TRIGﾌﾗｸﾞをｾｯﾄ

        movlw  1<<HALF          ; HALFﾌﾗｸﾞを反転する
        xorwf  stat,F           ;

int_90:
        bcf    INTCON,T0IF      ; ﾀｲﾏ割り込みﾌﾗｸﾞをｸﾘｱ
        pop                     ; ｺﾝﾃｷｽﾄ復元
        retfie                  ; ﾀｲﾏ割り込み処理終了

;======== 初期設定 ======================================
initialize:
        clrf   PORTA            ;
        clrf   PORTB            ;

        bsf    STATUS,RP0       ;

        movlw  b'00000000'      ; PORTAの入出力方向を設定
        movwf  TRISA            ;

        movlw  b'10000000'      ; PORTBの入出力方向を設定
        movwf  TRISB            ;

        movlw  b'00010001'      ; TMR0の設定
        movwf  OPTION_REG       ;  (4命令ｻｲｸﾙごとに++)

        bcf    STATUS,RP0       ;

        movlw  0x50-(temp+1) ;zero clear 0x50 ~ 0x0d
        movwf  temp             ;
        movlw  (temp+1)         ;
        movwf  FSR              ;
clr_lp:                         ;
        clrf   INDF             ;
        incf   FSR,F            ;
        decf   temp,F           ;
        btfss  STATUS,Z         ;
        goto   clr_lp           ;

        movlw  COUNT0_INIT      ; 時間計測ｶｳﾝﾀをﾘｾｯﾄ
        movwf  count0           ;

        bsf    INTCON,T0IE      ; ﾀｲﾏ割り込みを許可
        bsf    INTCON,GIE       ;

;======== メインルーチン ================================
main:
        btfss  event,TRIG       ; TRIGﾌﾗｸﾞが1なら以下へ
        goto   main             ; それ以外はmainへ

        btfsc  stat,HALF        ; HALF=0なら前半処理へ
        goto   main_10          ;
        call   first_half       ;
        goto   main_90          ;
main_10:                        ;
        call   second_half      ; それ以外は後半処理へ
main_90:
        bcf    event,TRIG       ; TRIGﾌﾗｸﾞをﾘｾｯﾄ
        goto   main             ; mainへ戻る

;======================================================
first_half:
        bcf    PORTA,STICK      ; ｿﾚﾉｲﾄﾞをOFF

        call   select_memory    ; FSR  <- step>>3
                                ; temp <- stepの下位3ﾋﾞｯﾄ

        btfss  INTCON,RBIF      ; 入力に変化あり？
        goto   first_half_10    ;     なければ10へ

        movf   temp,W           ; ﾘｽﾞﾑﾊﾟﾀｰﾝに音符を書き込む
        iorwf  INDF,F           ;
        goto   first_half_20    ;

first_half_10:
        comf   temp,W           ; ﾘｽﾞﾑﾊﾟﾀｰﾝに休符を書き込む
        andwf  INDF,F           ;
        goto   first_half_20    ;

first_half_20:
        movf   PORTB,F          ; RBIFのﾘｾｯﾄ
        bcf    INTCON,RBIF      ;

        incf   step,F           ; stepの更新
        movf   step,W           ; (0～STEP_MAX-1を繰り返す)
        sublw  STEP_MAX         ;
        btfsc  STATUS,Z         ;
        clrf   step             ;

        return                  ;

;======================================================
second_half:
        call   select_memory    ; FSR  <- step>>3
                                ; temp <- stepの下位3ﾋﾞｯﾄ

        movf   INDF,W           ; 音符があったらｿﾚﾉｲﾄﾞをONに
        andwf  temp,W           ;
        btfss  STATUS,Z         ;
        bsf    PORTA,STICK      ;

        return                  ;

;======================================================
; ｻﾌﾞﾙｰﾁﾝ : select_memory
; stepの値に応じてｱｸｾｽするﾒﾓﾘのｱﾄﾞﾚｽと、ﾋﾞｯﾄ位置を設定する
;        FSR  <- step >> 3
;        temp <- stepの下位3ﾋﾞｯﾄ
;======================================================
select_memory:
        rrf    step,W           ; 現在のstepに対応する
        movwf  temp             ;     音符が格納されている
        rrf    temp,F           ;          ｱﾄﾞﾚｽを算出
        rrf    temp,W           ; (FSR <- step >> 3)
        andlw  b'00011111'      ;
        addlw  rhythm_buf       ;
        movwf  FSR              ;

        movf   step,W           ; 現在のstepに対応する
        andlw  b'00000111'      ;     音符が格納されている
        call   bit_table        ;          ﾋﾞｯﾄ位置を指定
        movwf  temp             ; (temp <- stepの下位3ﾋﾞｯﾄ)

        return

bit_table:                      ; 0～7の数値を
        addwf  PCL,F            ;      ﾋﾞｯﾄ位置に変換するﾃｰﾌﾞﾙ
        retlw  1<<7             ;
        retlw  1<<6             ;
        retlw  1<<5             ;
        retlw  1<<4             ;
        retlw  1<<3             ;
        retlw  1<<2             ;
        retlw  1<<1             ;
        retlw  1<<0             ;

;======================================================
        end
```


参考文献

山名宏治
作りながら学ぶロボット工作入門⑤
トランジスタ技術
2001年5月号pp.155～160
CQ出版（株）

Linux環境でプログラミングを学ぼう！

よいこのLinux修行

第6回 CGIを使ってみよう

横井 勝 <masaru@turbolinux.co.jp>

今回は，多くのWebサイトで利用されているCGIについて紹介します。

1. CGIとは

CGI（Common Gateway Interface）はプログラムをWebサーバで実行して，その結果をWebサイトを見ているクライアントに渡すために，そのプログラムを実行する環境のことをいいます。

Webサーバなどでコンテンツを公開する場合，HTMLなどの形式で書かれたデータをファイルという形式で公開する方法が一般的に使われています。この場合，Webサーバは，指定されたファイルの内容を読んでクライアントに送るという動作をしています。それに対し，CGIではファイルでなくプログラムの実行結果をクライアントに返すようになっています。

CGIはWebサイトの閲覧者数をカウントしている「アクセスカウンタ」や，Web上でのBBSやチャットのシステムなど，さまざまな場所で利用されています。

CGIで実行するプログラムのなかで，Perlなどで実行するスクリプトのことを「CGIスクリプト」と呼ぶ場合があります。

1.1. CGIに使えるプログラム

「CGIスクリプト」と聞くと，perlで動くスクリプトを連想される人も多いかと思います。しかしCGIはPerlに限らず，C言語などで書かれたソースをコンパイルしてできたバイナリや，シェルのスクリプト，RubyやPythonやJavaで実行できるプログラムなど，その場でコンピュータが実行できる形式ならCGIとして利用できます。

2. ApacheによるCGIの利用方法

2.1. Apacheの準備

TurbolinuxにはWebサーバのApacheというソフトウエアが準備されています。インストールタイプによってはインストールされていない場合もあります。インストールされていない場合は，TurbolinuxのCD-ROMの中のapacheのパッケージをインストールするか，TurbolinuxのFTPサーバから入手してインストールしてください。

今回は，Turbolinuxが配布しているApacheのパッケージを利用することを仮定した設定方法などを紹介します。他の方法を利用する場合，ディレクトリの場所などが異なる場合があります。

インストール直後のApacheの設定では，URLのhttp://localhost/cgi-bin/にあたる/home/httpd/cgi-bin/(*1)以下のディレクトリにあるすべてのファイルがCGIのプログラムとして設定されています。その反面，このディレクトリ以外ではCGIを利用することができません。

CGIのテストなどを，すべてrootが行うなら構わないのですが，ある程度，ログインしたユーザーの権限で設定を行えると便利でしょう。各ユーザーがCGIを利用できるようにするには，設定ファイルの変更が必要になります。

一般ユーザーもCGIが利用できるようにするための設定を紹介します。ここからの作業はユーザーrootになって行ってください。

まずは，Apacheの設定ファイルである/etc/httpd/conf/httpd.confのなかにある

```
#AddHandler cgi-script.cgi
```

という部分の「#」を消してください（エディタviを使っているときには，「/AddHandler [Enter]」とタイプするとこの部分を簡単に見つけることができます）。この設定を行うと，ファイル名が.cgiで終わるファイルはCGIのプログラムとしてApacheは扱います。

そして，各ユーザーのホームディレクトリの下のディレクトリ"public_html"でもCGIが利用できるように「<Directory/home/＊/public_html>」から「</Directory>」の部分の行頭にある「#」を消して，「Options」の部分にExecCGIという項目を追加してください（**第1図**）。

この設定によって，各ユーザーはホームディレクトリの下にpublic_htmlという名前のディレクトリの中にあるファイ

```
<Directory /home/*/public_html>
    AllowOverride FileInfo AuthConfig Limit
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec ExecCGI
    <Limit GET POST OPTIONS PROPFIND>
        Order allow,deny
        Allow from all
    </Limit>
    <LimitExcept GET POST OPTIONS PROPFIND>
        Order deny,allow
        Deny from all
    </LimitExcept>
</Directory>
```

《第1図》httpd.confの変更箇所

（＊1）/home/httpd/cgi-binについて＝つぎにリリースされるTurbolinuxでは/home/httpd/cgi-binは/var/www/cgi-binに変更される予定です。

ルをCGIのプログラムとして実行することができるようになります。

これらの設定の作業が終わったら，Apacheのリスタートをするために，以下のようにコマンドを実行してください。

```
/etc/rc.d/init.d/httpd restart
```

システム起動時から Apache を実行したい場合は，以下のコマンドを実行してください。

```
/sbin/chkconfig --add httpd
/sbin/chkconfig httpd on
```

2.2. テストしてみる

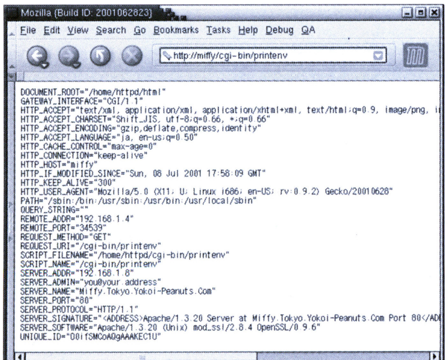
Apacheには，ディレクトリ /home/httpd/cgi-bin/ の中には，printenv と test-cgi というファイル名の二つの簡単なCGI スクリプトが準備されています。これらを実際に動かしてみましょう。

まずはrootになって，上記の二つのスクリプトの実行権限を設定します。

```
# cd /home/httpd/cgi-bin/
# chmod +x printenv
# chmod +x test-cgi
```

それではWebブラウザでこのスクリプトの実行結果を見てみましょう。たとえばWebサーバと同じコンピュータから見る場合，URLに「http://localhost/cgi-bin/printenv」と指定してみてください（**画面1**）。Webサーバにしているコンピュータで X Window System が動いていなくても，lynxやw3mなどのテキストブラウザを利用すれば，ターミナルでWeb サイトを閲覧できます。

もちろん，他のコンピュータとEthernetでTCP/IPによる通信が行えるなら，他のコンピュータのWebブラウザから閲覧できます。その場合は，URL の部分にWebサーバになっているホスト名またはIP アドレスを指定してください。



《画面1》printenvの実行結果の例

3. BBS のプログラム

CGIを利用した簡単なプログラムの例として，簡素な掲示板のプログラムをPerlで書いてみました（**リスト1**）。この掲示板のプログラムに装備されている機能は，メッセージのファイルを読み込む機能と，メッセージのファイルにメッセージを書き込む機能だけです。複数のユーザーが同時に書き込もうとしたときの衝突を防ぐためのファイルのロックなどの機能は装備されていません。ファイル名はboard. cgi になっています。

3.1. 使いかた

BBSのプログラムであるboard.cgiをインストールする場所として，/var/httpd/cgi-bin/の下か，ユーザーのホームディレクトリの下にpublic_htmlというディレクトリを作成して，その下に置く場合の２種類があります。

/home/httpd/cgi-bin/board.cgi にこのプログラムを置く場合，rootになって以下の作業を行ってください。

```
# cd /home/httpd/cgi-bin/
# cp [どこか]/board.cgi ./
# chmod 755 board.cgi
# touch mes.txt
# chmod 666 mes.txt
```

この作業にある「[どこか]」というのは，board.cgi のあるディレクトリのことです。最後のほうで「touch mes.txt」と「chmod 666 mes.txt」とやっているのは，メッセージを書き込むためのファイルを準備するためです。ここまでの作業が終わったら，WebブラウザのURLに

```
http:// [ホスト名]/cgi-bin/
board.cgi
```

と指定して実行してみてください。「[ホスト名]」の部分はWeb サーバのホスト名になります。

それに対し，たとえば現在のユーザーのホームディレクトリの下の public_html/cgiというディレクトリにこのプログラムを置く場合，以下のようにしてください。

```
$ mkdir -p ~/public_html/cgi
$ cd ~/public_html/cgi/
$ cp [どこか]/board.cgi ./
$ chmod 755 board.cgi
$ touch mes.txt
$ chmod 666 mes.txt
```

そしてWeb ブラウザのURL に

```
http://[ホスト名]/~[ユーザー名]/
cgi/board.cgi
```

と指定して実行してみてください。「[ユーザー名]」の部分は，そのユーザーの名前になります。

このプログラムは，日本語の文字コードセットの変換にjcode.plというPerlのライブラリを用いています。jcode.plはftp://ftp. iij.ad.jp/pub/IIJ/dist/utashiro/perl/ から入手できます。もし，jcode.plがインストールされていない場合は，board.cgi と同じディレクトリにjcode.pl をコピーしてください。

jcode.plがインストールされているかどうかをチェックするには，シェルで以下のように実行してください。何もメッセージが表示されなければインストールされています。

```
$ perl -e "require 'jcode.pl'"
```

試しに，このboard.cgi を実行してみましょう。board.cgiのあるディレクトリに移動して以下のように実行してみてください。

```
$ ./board.cgi
```

3.2. 動いた（?）動かなかった（?）

今回の記事のとおりにやっても，うまくいかない場合があるかもしれません。そんなときには，このBBSのプログラムをシェルで直接実行してみたり，Apache のエラーログのファイルである /var/log/http/error_log などを参考にしてみてください。

```
#!/usr/bin/perl

require 'jcode.pl';

# 表示するメッセージの件数
$MAXMES = 50;
# メッセージを保存するファイル
$MESFILE = "./mes.txt";
# デバッグする場合は、$DEBUG = 1 を設定してください。
#$DEBUG = 1;


if ($ENV{'REQUEST_METHOD'} eq 'POST') {
    # <form method="POST"> の場合
    &writemes;
} else {
    &printmes;
}

sub writemes {
    ## メッセージの書き込み
    # form で送られた文字列を $qs に代入する。
    read(STDIN, $qs, $ENV{'CONTENT_LENGTH'});
    @a = split(/&/, $qs);
    foreach $each (@a) {
 if ($each =~ /^(.+)=(.+)$/) {
     $qi ++;
     ($name, $val) = ($1, $2);
     $val =~ tr/+/ /;
     $val =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack('C', hex($1))/eg;
     $val = jcode::euc($val);
     $val =~ s/&/&amp;/g;
     $val =~ s/</&lt;/g;
     $val =~ s/>/&gt;/g;
     $val =~ s/¥"/&quot;/g;
     $Query{$name} = $val;
 }
    }
    if ($Query{'name'} eq "") {
 # 名前が指定されていない場合は "名無し" にする。
 $Name = "名無し";
    } else {
 $Name = $Query{'name'};
 # 改行をさせないようにする。
 $Name =~ s/¥n//g;
 $Name =~ s/¥r//g;
    }
    if (length($Query{'body'}) == 0) {
 # 本文が書かれていないときには エラーになる。
 print "Content-Type: text/plain; charset=EUC-JP¥n¥n";
 print "前のページに戻ってから、本文を書いて下さい。m(..)m¥n";
 return();
    } else {
 $Body = $Query{'body'};
 # 改行を <br> タグに書き換える。
 $Body =~ s/¥r¥n/¥n/g;
 $Body =~ s/¥r/¥n/g;
 $Body =~ s/¥n/<br>/g;
    }
    $datestr = localtime;
    $Mes = sprintf("<dt>%s(%s)</dt><dd><pre>%s</pre></dd>¥n",
     $Name,$datestr, $Body);
    if (open (IN, $MESFILE)) {
 $i = 0;
 while (($tmp = <IN>) && ($i < $MAXMES)) {
     $Mes .= $tmp;
     $i ++;
 }
 close (IN);
    }
    if ( ! open (OUT, "> $MESFILE")) {
 print "Content-Type: text/html; charset=EUC-JP¥n¥n";
 print "メッセージデータのファイルに書き込みができません。m(..)m¥n";
 return();
    }
    print (OUT $Mes);
    close (OUT);
    #print "Content-Type: text/html; charset=EUC-JP¥n";
    print <<__END__;
Content-Type: text/html; charset=EUC-JP

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
<head>
<title>メッセージを書きました。</title>
<meta http-equiv="Content-Type" content="Text/HTML; charset=EUC-JP">
<style type="text/css">
<!--
    H1 {color: blue;}
    H2 {color: green;}
    DT {color: red;}
//-->
</style>
</head>
<body>
<h1>メッセージを書きました。</h1>

<p>
ブラウザの再読み込みボタン を押さないでください。
<p>
[<a href="$ENV{'REQUEST_URI'}">メッセージをみる</a>]
</body>
</html>
__END__
}

sub printmes {
    print <<__END__;
Content-type: text/html; charset=EUC-JP


<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
<head>
<title>board.cgi</title>
<meta http-equiv="Content-Type" content="Text/HTML; charset=EUC-JP">
<style type="text/css">
<!--
    H1 {color: blue;}
    H2 {color: green;}
    DT {color: red;}
//-->
</style>
</head>
<body>
<h1>board.cgi</h1>
<h2>メッセージを書く</h2>
<p>
<form action="$ENV{'REQUEST_URI'}" method="POST">
<table border="1">
<tr>
<td>おなまえ</td>
<td><input type="text" name="name" size=20></td>
</tr>
<tr>
<td colspan=2>
<textarea name="body" cols=60 rows=5>
</textarea>
</td></tr>
<tr><td colspan=2 align="center">
<input type="submit">
<input type="reset">
</table>
</form>
<h2>メッセージ</h2>
__END__

    if (!open(IN, $MESFILE)) {
 print "<p>メッセージファイルが開けません。m(..)m¥n";
    } else {
 print "<dl>¥n";
 print while (<IN>);
 print "</dl>¥n";
    }
    print ("</body>¥n</html>¥n");
}
```



《画面2》BBSのプログラムを実行した場合

初心者でもかんたんラクラク作れちゃう！！

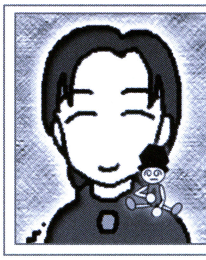


iモード・ホームページ講座

第6回　iモードホームページの今をつかむ

iモードで見られるカッコいいホームページ作りをめざしてきた講座も今号でいよいよ最終回。最後はホームページや電子メールなどのネット社会をめぐるすべてを総まくりしていきたいと思います。とくにこれからホームページを作り，運営していく私たちに必要なものってなんでしょう？　そのへんをマスコットキャラ·TAG君と語り合ってみました。

TEXT●白木ミホ



(左)東京産の神奈川育ち。iモード関連の執筆が多い。1年後にこの文を読めば「当時はこんなことしかできなかったのか」とビックリするくらい，劇的に進歩し続けるであろうiモードの世界ですが，「こんな頃からずっと続けているんだよ」といばれるくらい，皆さんの作品が長く続くことを祈っております。

(右)陰のサポーターTAG君。「/TAG」という双子の弟がいるらしい。ふたりでひとつ。

TAG君，ミホと語る

ミホ： ところで，TAG君はもう自分のホームページ立ち上げたのかな？

TAG君： もちろん！　まだ。

ミホ： あのね～。せっかく私と一緒にこの講座を盛り上げてくれてて，それはないんじゃない？

TAG君： いや～イザとなるとどんな内容にしようか，意外と悩んでさ。

ミホ： まずは日記とか，できるところから始めて，コンテンツはだんだん増やしていけばいいんだよ。

TAG君： なるほどね。がんばってみるよ。とりあえず，日記かな。

ミホ： そうそう，要はやる気だからね！

htmlの書きかたのコツは？

TAG君： ところで，この講座を読んでhtmlタグの意味とか使いかたは分かったんだけどさ，実際にリスト化するときのタグの書きかたってどれがホントなの？　改行とか段落下げとかいろいろあるみたいなんだけど。

ミホ： 確かにhtmlタグは人によっていろんな書きかたをしてるけど，どれも「見やすく」「キレイに」がモットーだね。作業する人がいかに分かりやすいかがミソで，自分なりの書きかたを開発してる人もいるよ（**第1図**）。

TAG君： 一般のプログラミングとおんなじだね。やっぱり意外と個性って出るんだ。

ミホ： そうそう。“
”とか“<p>”とか，画面上のレイアウト指示はタグで書くんだから，極端な話，タグのリストは1行で書ききってもいいんだよね。

TAG君： それは見にくいって。

友達に自慢したくなる爆笑サイトを教えて

TAG君： でも，自分でホームページを作るとネットサーフィンがさらに面白くなるよね。

ミホ： それだけでも大いなる進歩だよ。

TAG君： 普段どんなサイトを見てるの？　なんか面白いサイトはないかな？

ミホ： それを自分で作れよ。ま，確かに面白いサイトを見つけると友達に自慢したくなるけどね。

TAG君： 自慢してして。

ミホ： じゃあこんなのどう？　『いきなり次回予告』。前回取り上げたCGIを使ったホームページなんだけど，名前を三つ入力する（**写真1**）とCGIが勝手に予告編を作ってくれる（**写真2**）の。偶然の妙とはいえ爆笑ものだよ。

TAG君： 表示が文字だけだからどの機種でも楽しめるね。でも俺、こないだとうとう念願のカラーiモード買ったんだ。せっかくだからこのキレイな画面をフルに楽しめるサイトってないかな。

ミホ： それだったら『ウルトラ100兄弟（**写真3**）』がオススメだね。これは色が見

htmlタグの書きかたいろいろ

```
<html>
<head>
<title>史上最悪の<br>乗り物･ﾗｸﾀﾞ</title>
</head>
<body>
<center><IMG src="cattle1.gif"></center>
<p>
  四足歩行の常識を覆す，それがﾗｸﾀﾞだ。
  何しろ奴らは右前足と右後足を同時に前へ出す唯一の例外的生物なのだ。
  人間で言えば右手と右足がいっぺんに出ているようなもの。
  その歩き方と砂漠の環境に何か関係はあるのか?<br>
</p>
<a href="../index.html">ﾄｯﾌﾟﾍﾟｰｼﾞはこちら</a>
</body>
</html>
```

タグのとじ忘れがあればすぐ分かる

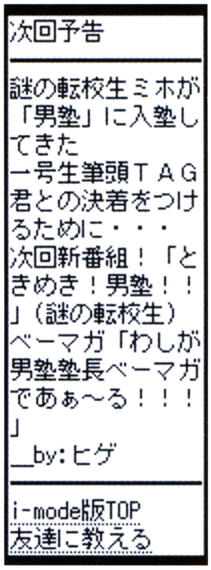
```
<html>
<head>
<title>
      史上最悪の
<br>
      乗り物･ﾗｸﾀﾞ
</title>
</head>
<body>
<center>
      <IMG src="cattle1.gif">
</center>
<p>
      四足歩行の常識を覆す，それがﾗｸﾀﾞだ。
      何しろ奴らは右前足と右後足を同時に前へ出す唯一の例外的生物なのだ。
      人間で言えば右手と右足がいっぺんに出ているようなもの。
      その歩き方と砂漠の環境に何か関係はあるのか?
<br>
</p>
<a href="../index.html">
      ﾄｯﾌﾟﾍﾟｰｼﾞはこちら
</a>
</body>
</html>
```

修正場所がささっと書きかえられる

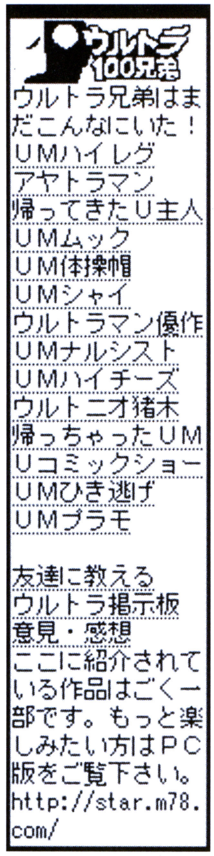
《**第1図**》上は一つのタグに1行をモットーに段落ふうに仕上げてある。そして，下はテキストと図の記述部分を下げてタグを見やすくしてある



《写真1》CGIでランダムに文章を作ってくれる。試しに「TAG君」「ミホ」「ベーマガ」と，三つの名前を入力すると…



《写真2》こんな「次回予告」ができて大爆笑



《写真4》ホームページを見に行く必要のない，メール配信型サービス。いつでも欲しい情報のジャンルを変更できるのもうれしい

《写真3》ウルトラマンが国民的芸術になる日も近いか。パソコンをお持ちの方はぜひ投稿を

られないと楽しめないからね。講座の第4回で取り上げた画像の技術と，参加型ホームページとしての面白さをいちばん楽しめるよココは。

TAG君：笑い過ぎで腹イタイ～。

オトクなサイトをチェック

ミホ：TAG君は携帯端末独自の「印籠方式」は利用してる？

TAG君：それは「画面を見せれば割引します」ってサービスのこと？

ミホ：そう。あれはホントに水戸黄門の印籠だよね。ええいっ，この画面が目に入らぬかってね。

TAG君：まあ，ユーザーとしてはキモチのいい瞬間ではあるよね。

ミホ：でしょう？　代表格はやっぱりグルメの検索サイト『ぐるなび』で決まりでしょう。無料の公式サイトってのがまたオトク感倍増。ああ，一度でいいから高級レストランのギャルソン相手にやってみたい。

TAG君：グルメの他にはそういう割引系はないの？

ミホ：ｉモードで受信したメールで割引や優待が受けられる『@DO（第4図）』っていうクーポンの総合サービスがあるよ。ホームページで登録すれば，ポイントごとに商品券のキャッシュバックがあるし，お誕生月には抽選でプレゼントが当たるBirthdayキャンペーンとかもあるし。

TAG君：いろんなジャンルを回る手間が省けて，なおかつ登録無料ってところは時間的にもオトクだね。

ミホ：トクってことから逆に発想すれば，「損をしないためのサイト」も押さえといたほうがいいよ。

TAG君：節約ってこと？

ミホ：ううん，もっと深刻な，壊滅的な打撃の話。たとえばサイフを落としちゃったとか，免許証やキャッシュカードをなくしちゃった，なんて経験ない？

TAG君：あるある。カードとかは悪用されないように，銀行に電話してすぐ止めなきゃいけないってことは分かってたけど，イザとなるとどうしたらいいか分からなくって。

ミホ：そういう緊急時に役立つ，心強い電話番号ばかりを集めたサイトってのが『ドコモ情報ダイヤル』なんだ。これも公式サイトなんだよ。

TAG君：へえ、こんなのもあるんだね。マイメニューからイッパツで呼び出せればパニクってても大丈夫そう。

「リング」ってなに？

TAG君：ところで，いろんな情報はどこからゲットしてくるの？

ミホ：もちろん地道なネットサーフィンの賜物よ。でも最近は検索サイトもたくさんあるし，リングもあるから調べやすいよ。

TAG君：あっ，それパソコン向けホームページで最近よく見るよ。リングって何？

ミホ：類は友を呼ぶ。同じ趣味や傾向を共有するホームページのリンク集なんだ。このリングに参加しているホームページを次から次へとたどって行けるようにリンクが張ってあって，それがひとつなぎの輪（リング）になってるの。まだ見ぬサイトを開拓するには便利だよ。ｉモード版はまだ少ないから，立ち上げるなら今！だね。

ユーザーはつらいよ 迷惑メール編

ミホ：TAG君，こないだ迷惑メールが来たって言ってたけど，最近どう？

TAG君：ミホの言うとおり，メールアドレスを変更したから，今は大丈夫。

ミホ：端末を買ったときに自動的についてくる電話番号の仮アドレス(携帯の番号@docomo.ne.jp)をそのまま使ってたからね。

TAG君：迷惑メールといえば，少し前にCMで見たよ。迷惑メール100通分の通信料を無料にしますっていうの。

ミホ：8月1日から実施の新サービスね。ｉモードは情報料を受信者が払う仕組みだから，迷惑メールの負担はホントに痛かったよね。

TAG君：迷惑メールってなんとか撃退できないのかな。

ミホ：まず，電話番号のアドレスを使ってる人は即変更することだね。これも7月8日契約分からは初期設定がランダムな英数字の組み合わせになるから，今後は改善されるわけだけどね。

TAG君：なるほど。あと不思議なのは，俺の女友達のところに最近やたらと出会いサイトの勧誘メールが来てるんだけど，それはなんで？

ミホ：最近，ホームページの掲示板とかにメールアドレスを書きこんだ？

TAG君：あちこち見てるみたいだけど。

ミホ：とくにそのアドレスがhanakoとかyuriみたいな，明らかに女性の名前だったりすると，エッチ系に狙われるケースはあるよ。
TAG君：なるほど〜。
ミホ：たとえ健全なサイトでも狙うヤツはいるからね。女性はとくに注意しないといけないね。
TAG君：あと友達のとこに「このメールを7人に送信しないとパケット料を全額負担させられます」みたいなメールが来たんだって。コワイよな。
ミホ：そんなの大ウソだよ。ゼッタイありえないことなんだから，送信なんかして悪事の片棒かついじゃダメだよ。
TAG君：でも知らないとホントにコワイよね。

ユーザーはつらいよ 掲示板編

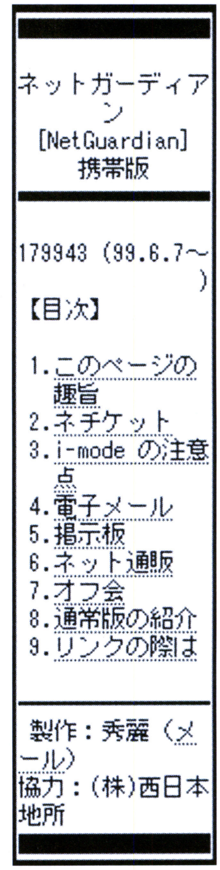
ミホ：でも最近特に被害甚大なのは，掲示板にくる「アラシ」だね。
TAG君：アラシって何？
ミホ：誰でも自由に書きこめる機能を悪用して，中傷文を書いたり，大量の書きこみで掲示板のページを埋め尽くしたりする妨害行為なんだ。不快なのはもちろんだけど，掲示板そのものが成り立たなくなって閉鎖に追いこまれたりするんだ。
TAG君：何が楽しくてそんなことするんだろ。
ミホ：誰かが自分に反応してくれることを期待してるんだね。人を怒らせて楽しんでるんだ。
TAG君：へんなの。
ミホ：でも多いんだよ，そういう書きこみが。
TAG君：自分のホームページにそういう人が来たらどうすればいいの？
ミホ：削除！　そして無視無視無視ッ！見ても決して反応しちゃいけないよ。「やめてください」とか，「なんでこんなことするんですか？」なんて書いても，アラシを喜ばせるだけだからね。
TAG君：参加してる人たちにも注意がいるね。
ミホ：そのとおり。管理人と参加者が一丸となって無視すること。消極的に聞こえるかもしれないけど，それがいちばん大事なんだよ。

恐怖のネット犯罪

TAG君：アラシって結局はネット犯罪だよね。
ミホ：もちろん！
TAG君：最近多いみたいだけど，他にどんなのがあるの？
ミホ：いちばん怖いのは個人情報を流すタイプね。電話番号や住所を勝手に掲示板に書かれちゃったら，いくら削除しても情報そのものは取り返せないからね。なるべく個人情報はネットに流さない，これ鉄則。
TAG君：自分から書いてる人を見たことあるよ。
ミホ：でもたとえばTAG君が「こんにちはぁ♪　私ミホ。XXXX-XXXXまでお電話ちょうだいね」なんて書きこみしても，見てる人には分からないじゃない。いわゆる他人を装った悪質な書きこみかも。そういうのは管理人に連絡してすぐ削除してもらうべきね。
TAG君：なるほど。でも掲示板で仲良くなった人とかと直接連絡を取りたいときは？
ミホ：それも注意してね。ネカマって知ってる？
TAG君：あっ，知ってる。ネットおかまでしょ？　ネット上で女のフリしてる男。
ミホ：そう。たとえば女性同士だからって気安く連絡先を教えたら，相手がとんでもないストーカー男で自宅に押しかけられた，なんて事件は残念ながらよくあるのよ。TAG君が男でも要注意。女だったら厳重注意よ。
TAG君：そういうヤツらにとって，ネットって天国なんだろうな。
ミホ：その温床はネットの持つ匿名性なんだよね。それだけの被害にあっても，加害者の住所はおろか，フルネームさえ知らない，なんてことはホントに多いから，そのへんが今までの犯罪とは大きく違うところね。

iモードユーザーの深い憂鬱 ネチケット

TAG君：コワイ話だなあ。
ミホ：まあ最悪の事態ってことね。でもたとえ犯罪とは言いきれない範囲でも，ネット上で不愉快な思いをすることってあるよね。
TAG君：あるある。こないだ俺が書きこみした掲示板の記事にぜんぜん知らないヤツが，「TAG，へこんでるじゃ〜ん？」とかタメぐちでレス(返信)つけてきてさあ。
ミホ：それは明らかなネチケット違反だ。
TAG君：さらに地域限定発売のお菓子の話で盛りあがったときには，ぜんぜん知らないヤツが「それ欲しいから買って送ってください」とか言うんだよ？　確かに犯罪じゃあないけど，それはないだろって激論になっちゃったよ。



《写真5》自分だけは大丈夫，なんて気持ちがいちばんアブナイ。正しいユーザーなら必見

ミホ：そういう明らかに困ったちゃんな場合は，みんながTAG君の味方について反撃してくれると思うけど，タメ口とかだと「私は親しみがあっていいと思います」なんてズッコケたことを言う人も出かねないよね。繰り返すけどそれもネチケット違反だからね。
TAG君：もしかしたら俺も，知らないうちに誰かに不愉快な思いをさせてるのかな。
ミホ：そういう謙虚な姿勢が大事だよ。知らなかったゴメンナサイって言い訳は，私はあまり好きじゃないよ。iモードはじめ携帯端末のユーザーには特に知ってほしいことだね。
TAG君：なんで携帯って限定するの？
ミホ：知らないの？　今，携帯端末ユーザーはネチケット知らずで一般のパソコンユーザーから大ヒンシュク買ってるんだよ。

TAG君：マジーッ？

ミホ：携帯やPHSの普及で電子メールが身近になった分，ネチケットとかを知らずにネット社会に入って来ちゃう人が爆発的に増えてるせいなの。少し前まではメール＝パソコン＝社会人の使うものだったけど，今は小学生でもネットに参加できるでしょ？　たとえばさっきの言葉遣いの問題でも，面と向かって会う初対面の大人にいきなりタメ口使う小学生はいないけど，ネット社会だと目に入るのはパソコンの画面だけだから，それが分からなくなっちゃうのね。でも子供だからって許されないのがネット社会でもあるわけ。

TAG君：じゃあ，ルールはどこで知ればいいんだろう。

ミホ：たとえば『ネットガーディアン（写真5）』では，メールや掲示板でのネチケットから，パソコン仲間と直接会うオフ会の心得や，万一被害に遭ったときの対処法まで，「自分を守り」「他人を傷つけない」ための情報が書かれてるよ。ネチケット情報を扱うホームページはたくさんあるから，情報は積極的に手に入れたいね。

ホームページ管理者の心得

TAG君：自分のホームページを作るってことは，htmlタグを覚えたりするだけじゃないんだね。

ミホ：作ることよりも管理すること，継続することのほうがじつは大変で，しかも重要なんだよね。

TAG君：俺のページはどうなるのかなあ。

ミホ：コワイことにならないためには，まず毅然とした態度で対応すること。これにつきるね。あとは管理人さんがつねに顔を出す。管理者不在のページはさびれるし，荒れるよ。ちょっとずつでもマメな更新をして，楽しめるホームページを作ろう！　っていう熱意が伝わることが，良いホームページの極意だね。

TAG君：作るほうも，見るほうも楽しくなくっちゃね。

ミホ：そういうページには自然といい人が集まるから，アラシなんかが来てもたくさんの常連さんが管理人さんを助けてくれるよ。

TAG君：自分が出入りしているホームページでも，主催者を助けていくくらいの気持ちがないといけないんだね。

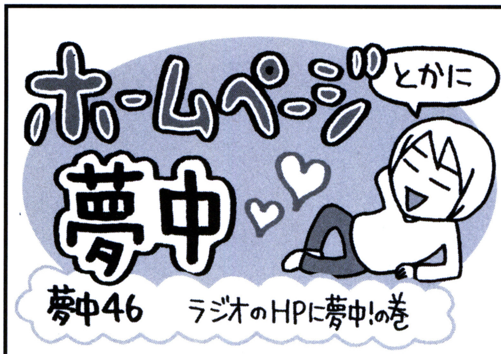
ミホ：そう！　それが大事。良いホームページは良い管理人がいることが絶対条件だけど，同時に管理人1人ではできないこともあるんだよ。がんばってね。



《写真6》TAG君のサイト。最初は少しずつ更新できる範囲でコンテンツを作ろう！

今回紹介したサイト

いきなり次回予告	http://coc.coco.co.jp/i/
ウルトラ100兄弟	http://star.m78.com/i/index.htm
@DO	http://atmarkdo.net/
ネットガーディアン	http://guardian.ne.jp/i/



何か得できるホームページはないかなーと探していたところ辿りついたのがラジオ局のページ。

私は昼間東京ＦＭを聞きながらお仕事しているのですが，誰でも一つはよく聞くラジオ番組があるはず！

で，何がお得かというと，まず聞き逃したことを改めてチェックできること。試写会とかイベントの情報で宛先を聞き逃してもホームページで確認して送ることができるし，FAX番号を聞き逃しても，ホームページにあるフォームから直接リクエストを送ることができるのだ。

また，ＤＪがどんな様子で話してるのかも画像や映像でチェックすることができる！　とにかく聞くだけでなく目でも楽しむこともできるのだ。

ハマッてるラジオがあれば，ぜひチェックしてみよう！かなり楽しめること請け合いです。

（神奈川）



DTM Art Studio

パソコンで音楽をたのしもう！

協　力
ヤマハ（http://www.yamaha.co.jp）

Text&SoundData
仲野方敏／上村学

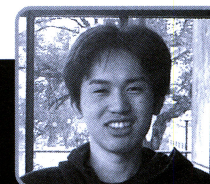
このコーナーは，芸大生が体験したDTMの基本，応用，実践を紹介するコーナーです

筆者紹介



仲野方敏：
大阪芸大
芸術学部　音楽学科
音楽工学コース（音楽工学専攻）3回生

いや～暑いですね。みなさんお元気ですか？大川・幸田に代わり，今回から4回にわたって記事を担当させてもらう仲野です。みなさんがんばって作曲していますか？僕たちも，暑さに負けないで音楽やっています。ということで，今回もがんばっていきましょう。



上村学：
大阪芸大
芸術学部　音楽学科
音楽制作コース（作曲専攻）3回生

みなさんはじめまして！！今月から4回分を担当することになった上村です。よろしくお願いします。もう，「めちゃらく作曲名人」は使いこなせるようになったでしょうか？今月からはさらに進んだソフトに挑戦しよう！ということではりきっていっしょにがんばりましょう。

YAMAHA「らくらく作曲名人2」～セットアップ&基本操作編～

YAMAHA「らくらく作曲名人2」

「らくらく作曲名人2」は，音楽の専門知識がなくても，作曲・編曲等が簡単にできるソフトです。「めちゃらく作曲名人」とちがい，シーケンサとして使うこともできるので，DTM初心者以外の人にもお勧めです。それに，楽譜を制作することも可能なので，多くの人が楽しめるソフトです。

「らくらく作曲名人2」のインストール

skMeijinというフォルダの中からsetup.exeをダブル・クリックします。すると，セットアップの準備が開始されます。セットアップの準備が終わると，「YAMAHAらくらく作曲名人2 InstallShieldウィザードへようこそ」という画面が出ますので，［次へ］をクリックします。すると，「ユーザーサポート」の画面が出てきますので，［次へ］をクリックします。「インストール先の選択」の画面が出てきます。
（※以前のバージョンのらくらく作曲名人を使っている人は，同じフォルダに上書きしてインストールされるので，以前のバージョンのらくらく作曲名人はあらかじめアンインストールしておいたほうがいいでしょう。）

ここでも［次へ］をクリックしましょう（パソコン側がインストール先のフォルダを選択してくれますが，自分で選択したい場合は［参照］をクリックすると自分で選択できますよ）。するとインストールが開始します。

インストールが完了すると.MIDファイルをらくらく作曲名人2に関連付けるか聞いてきますので［はい］をクリッ

「らくらく作曲名人2」

製品仕様
- CD-ROM（「らくらく作曲名人2」，ソフトシンセサイザ「S-YXG50」，ヤマハCBXドライバ，MidRadioプレーヤ，用語集）
- 取扱説明書

動作環境
- 対応パソコン：NEC PC-98シリーズ，IBM-PC／AT並びにその完全な互換機
- 対応OS：Windows Me／98／95／NT4.0／2000
- CPU：Pentium-166MHz以上／Celeron以上（ソフトシンセサイザー同時使用の場合MMX Pentium-233MHz／Celeron以上推奨，Windows2000の場合PentiumⅡ300MHz以上推奨）
- 必要メモリ：32MB以上（64MB以上推奨）
- ハードディスク必要容量：40MB以上
- ディスプレイ：800×600ドット，24ビットTrue Color以上
- 音源：ソフトシンセサイザまたはXG対応MIDI音源機器
- 必要機器：マウス，ハードディスク，2倍速以上のCD-ROMドライブ



14,800円

クしておきましょう。また同様に.XWSファイルを関連付けるか聞いてきますので［はい］をクリックしておきましょう。

すると「YAMAHA CBX Driver」をインストールするか聞いてきます。外部音源を使用する人はインストールしておきましょう。ソフトシンセサイザを使う人はとくにインストールの必要はありません。つぎにセットアップ完了のメッセージが出るので［OK］をクリックします。最後にコンピュータを再起動するか聞いてきますが，続けてソフトシセサイザーをインストールするので「いいえ，あとでコンピュータを再起動します」を選択して［完了］をクリックします（外部音源を使用する人，またはソフトシンセサイザーS-YXG50をインストール済みの人は，ここで再起動を選んで再起動してください）。

YAMAHAソフトシンセサイザー(S-YXG50)のインストール

先ほどインストールした中に＊Syg50_＊というフォルダがあり，その中のsetup.exeをダブル・クリックします。すると「らくらく作曲名人2」をインストールしたのと同じようにセットアップの準備がはじまりますので「らくらく作曲名人2」をインストールしたのと同じ要領で表示を読みながら進めてください。［次へ］か［はい］をクリックしていればほとんど問題ないでしょう。

最後にコンピュータを再起動するか聞いてきますので今度は「はい，直ちにコンピュータを再起動します」を選んで再起動しましょう。

「らくらく作曲名人2」の起動＆設定

画面左下の［スタート］メニューから［プログラム］→［YAMAHAらくらく作曲名人2］→［らくらく作曲名人］を選択すると，らくらく作曲名人2が起動します。起動したらまず，音源の設定をしましょう。［設定］メニューをクリックして［音源設定］を選択します。ここでこのソフトで使用する音源の設定をします。

今回はYAMAHAソフトシンセサイザー（S-YXG50）を使用するので「ソフトシンセサイザーを使用する場合」をクリックして［詳細設定］をクリックします。MIDI Outを「YAMAHA SXG Driver」にしましょう。後は［OK］をクリックして音源設定画面でも［OK］をクリックして完了です。

音色，パターン，テンポ等簡単なEDIT

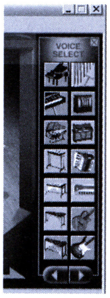
らくらく作曲名人では，めちゃらく作曲名人で作ったデータを編集することができます。編集できるのは，ボイス（音色），ホールタイプ（音響効果），スタイル（伴奏），テンポ等です。これらの編集は，専用のアイコンを表示して選択するだけなので，非常に簡単な操作で行うことができます。

ファイルの読み込み

まず「めちゃらく作曲名人」で作ったデータを読み込みます。［ファイル］メニューから［開く］を選択し，めちゃらくで作った.XWSか.MIDのデータを選んで［開く］をクリックします。

ボイスの編集

ボイスの編集を行うには，楽器パレットを開く必要があります。楽器パレットを開くには，ステージ画面右側のカーテン上にカーソルを移動させるか，｛表示｝メニューの中から｛楽器パレット｝を選択します。



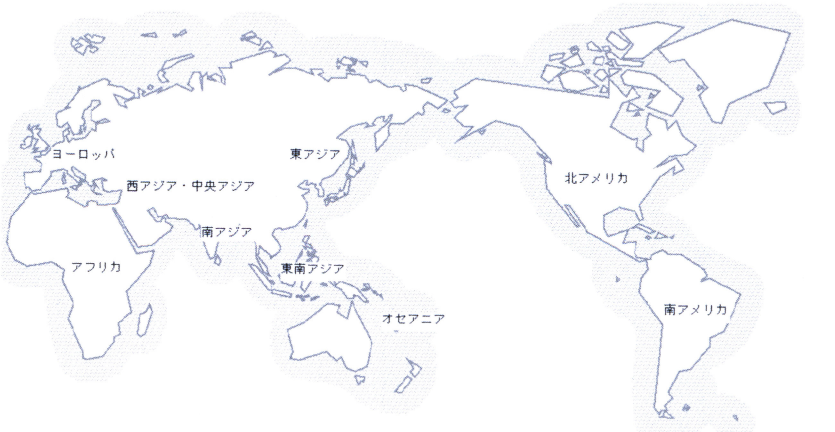
パレット上には，楽器アイコンが表示されます。この楽器アイコンをステージ上のプレーヤにドラッグ＆ドロップすることで，そのアイコンの楽器をプレーヤの楽器として設定することができます。また，ステージ上でいずれかのプレーヤが選択された状態であれば，楽器アイコンをクリックするだけで，そのアイコンの楽器を，選択されているプレーヤの楽器として設定することができます。

各楽器アイコンをマウス右ボタンでクリックすると，表示されるサブメニューの中から｛音色選択｝を選択する

民族音楽のスタイル

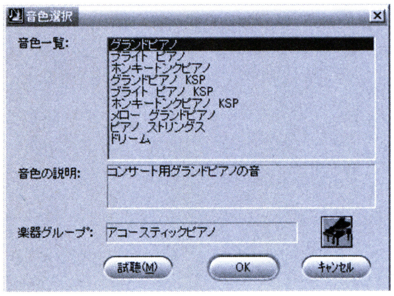
民族音楽と聞いてみなさんはどんなイメージを持つでしょうか？それは人によってちがいがあると思います。実際民族音楽は世界各地に存在していて，多くの人に知られていない音楽も存在しています。そこで，今回は民族音楽におおまかな位置付けをしたいと思います。

図のように世界の民族音楽は大きくわけることができると思います。アジアは一番広いから当たり前のような気がしますが，かなり多くの種類の民族音楽が存在していることがわかると思います。そして日本人の多くの人が民族音楽といえばアジアの音楽を想像するのではないでしょうか？そこでこの民族音楽のスタイルのコーナーではアジアの民族音楽を中心にいろいろな国や地域の民族音楽のスタイルを紹介していこうと思います。



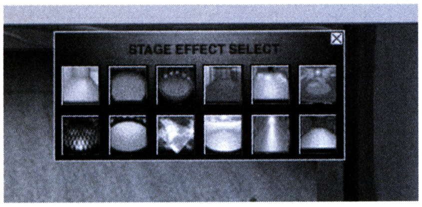
民族音楽のリズム等をDTM作品に取り入れたら新しい作風が生まれるかもしれませんよ。

と，音色選択画面が表示されます。この画面で，各楽器アイコンに割り当てられている楽器名を確認／変更することができます。各楽器アイコンに割り当てられている音色が気に入らなかったときは，ここで設定しなおすといいでしょう。



ホールタイプの編集

ホールタイプを編集するには，ステージパレットを開く必要があります。楽器パレットを開く場合とほぼ同じ要領で，ステージ画面右上の音響効果名（「HALL1」等と表示されている部分）の上にカーソルを移動させるか，｛表示｝メニューの中から｛ステージパレット｝を選択します。



ステージパレット上には，ステージアイコンが表示されます。いずれかのステージアイコンをクリックするか，ステージ上にドラッグ＆ドロップすると，ステージの風景が変化すると同時に，選んだアイコンに割り当てられている音響効果が設定されます。

アドバイス

ホールタイプを選ぶときは，曲の感じによって残響のタイプを変えるようにしましょう。たとえば，テンポの速い曲には比較的短い残響のホールタイプを。逆に，テンポの遅い曲（バラード等）には長い残響のホールタイプをというふうに，その曲によって変えるといいでしょう。

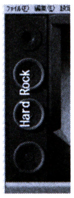
ただ，これはほんの一例で，速い曲だからといって絶対に残響時間を短くしなければいけないなどということではありません。要は，聴いていて最も良い（気持ちが良い）と思えるものを選べばいいのです。なんとも曖昧ですが，残響時間に関する定義（「…な感じの曲には…秒の残響を付加するのが最も良い」というようなもの）が存在しないので，このようにしか説明できないのです。

スタイルの編集

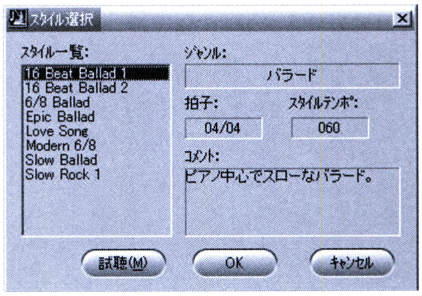
スタイルの編集についても，音色やホールタイプの編集と同様にパレットを開くことが必要になります。ステージ画面左側のカーテン上にカーソルを移動させるか，｛表示｝メニューの中から｛スタイルパレット｝を選択することでスタイルパレットが開きます。



パレット上にはスタイルアイコンが表示されます。いずれかのスタイルアイコンをクリックするか，ステージ上にドラッグ＆ドロップすると，ステージ両側のカーテンの形状が変化すると同時に，ステージ上のスタイル用のプレーヤが自動的に差し替わります。これで選択したアイコンに割り当てられているスタイルが演奏に有効となります。



また，各スタイルアイコンを右クリックすると表示されるサブメニューの中から｛スタイル選択｝を選ぶとスタイル選択画面が表示されます。



この画面で，各スタイルアイコンに現在割り当てられているスタイルを確認／変更することができます。これをうまく利用することにより，さまざまなスタイルを使用することができます。

シリーズ音響学

今回から始まりました「シリーズ音響学」。今回は，「音とは何か？」をテーマに進めたいと思います。

簡単に言うと「音」とは空気の振動です。難しい言いかたをすれば，"「音」とは弾力（弾性）と質量（慣性）を持った媒質(弾性体)の中を伝わる波動の一種"といえます。厳密には，空気中を伝わる疎密波(弾性波)ということになります。少し難しいかもしれませんが，何らかの振動が空気を押し，空気の圧力の平均（大気圧）より圧力の高い部分（密）と低い部分（疎）が交互に発生し，空気中を波として伝わっていくとイメージするとわかりやすいと思います。ちなみに，「音圧」とはこの圧力変化量のことです。

上記の媒質とは，音（波）が伝わっていくために必要なものであり，そのひとつに，私たちのまわりを取り巻いている空気があります。空気には，弾力（弾性）も質量（慣性）もあるので，音を伝えるための媒質となり得るわけです。ただ，ここであえて「媒質」としたのは，空気以外にも音を伝える媒質となり得るものが存在するからです。1方向に運動を続けようとする性質（慣性）とバネのように元に戻ろうとする性質（弾性）があれば振動が起こりうるので，他の気体や固体，もしくは液体でも音を伝えることができるのです。

しかし，媒質となる物質によって音の進む速さ（音速）は変わります。これは，物質の物理的特性が異なるためです。

それでは，最後に肩の力を抜いて読める話をしましょう。

真空状態では，媒質が存在しません。つまり，音が伝わらないのです。よく，宇宙を舞台にした映画で爆発音と共に宇宙船が粉々になるシーンを見かけますが，真空状態の宇宙空間では絶対にありえないことなのです。まあ，あの音がないとおもしろくも何ともないのはわかるのですが，ちょっと不思議な気分になってしまいます。

※補足

音速についてですが，一般には，1気圧，気温t（℃）の空気中では，

音速＝331.5＋0.6t（m／s）

とされています。つまり，気温15℃で約340m/s 。よって，気温15度の空気中では1秒間に340m進むということになります。

スタイルを楽譜に展開

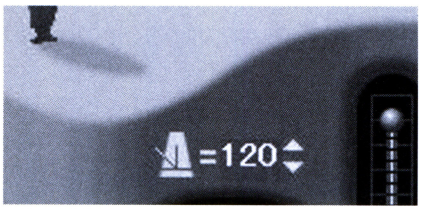
いずれかのスタイルが有効な状態で，{スタイルを楽譜に展開} アイコンをクリックすると，スタイル用のプレーヤがすべて通常のパート用のプレーヤと差し変わり，各プレーヤには，スタイルを現在のコードに基づいて展開した音符（MIDIデータ）が入ります。細か



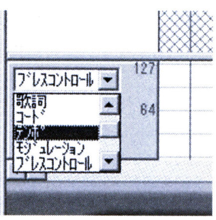
い部分を手直しする場合には，この機能を利用するといいでしょう。

テンポの編集

テンポのEDIT方法には，二つの方法があります。一つは，ステージ画面右下のメトロノームマークの右にある矢印をクリックしてテンポを変える方法。



ただ，この方法は一時的に再生速度を変えるだけのものなので，曲のデータとして記憶されません。テンポを曲のデータとして記憶させるためには，楽譜（またはピアノロール / ドラム）画面の左下にあるコントロールセレクト（ベロシティ，ピッチベンド，歌詞等と表示されている部分）をクリック



し，表示されるリストから「テンポ」を選択した後，その下にあるコントロールバリューで数値を設定します。

データの保存

データのEDITが終わったら保存しておきましょう。保存方法は［ファイル］メニューから［名前を付けて保存］をクリックして保存する場所を選択して保存する曲のファイル名を入力して保存する形式を選択して［保存］をクリックして保存完了です（保存できる形式はめちゃらく作曲名人と同じく.XWSと.MIDです）。

音符入力方法

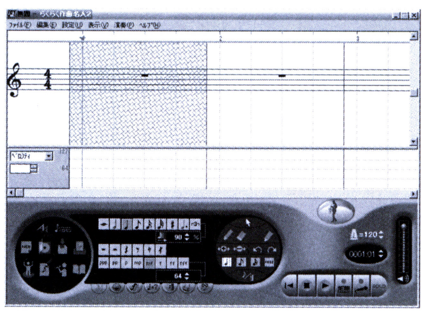
らくらく作曲名人はシーケンサとしても使うことができます。そこで楽譜（スタッフ）画面での音符入力方法を説明したいと思います。音楽について知識のある人はEDITしたデータにさらに自分で音符を入力してアレンジするとEDITデータ＋自分の音符入力でさらに個性的なアレンジもできます。

楽譜（スタッフ）画面開きかた

ステージ画面で足元の影が青色のプレーヤをダブル・クリックします。するとダブル・クリックしたプレーヤの演奏する楽譜（スタッフ）画面が出てきます。また，楽譜（スタッフ）画面でメトロノームの左上にあるステージ画面でダブル・クリックしたプレーヤのボタンをクリックするとステージ画面に戻ります。

楽譜（スタッフ）画面の説明

五線譜の画面は出てきましたか？それでは音符の入力の説明の前にこの画面のおおまかな説明をしましょう。



楽譜（スタッフ）画面の下半分の真ん中にある黒い枠の部分の下にあるボタンは，左から「音符パレット」,「音楽記号パレット」,「強弱記号パレット」,「速度記号パレット」,「反復記号パレット」,「奏法記号パレット」,「装飾記号パレット」を呼びだすためのボタンで，マウスでクリックしてやるとそのボタン上の真ん中の黒い枠部分にパレットが呼びだされます。これらのパレット



音符パレットとは？

ここでは主に音符入力に使われる音符パレットについて説明しましょう。音符パレットさえ使いこなせれば一応音符は簡単に入力できるようになっています。音符パレットでは音符の長さと強弱を選んで入力することができます。

音符の長さボタン

音符の長さの種類は音符パレット上の段左から順に「全音符」,「2分音符」,「4分音符」,「8分音符」,「16分音符」,「32分音符」と左から順に長い音符から短い音符になっていきます。

臨時記号ボタン

音符パレットの上段の32分音符のボタンの右の記号のボタンは音符に臨時記号をつける場合に使うボタンで，このボタンの上でマウスの右ボタンをクリックするとシャープ，フラット，ナチュラルに切り替えることができます。

付点ボタン

音符パレットの上の段の臨時記号のボタンの右の4分音符の横に点が付いたボタンは付点音符を入力するときに使います。

このボタンは先に紹介した音符の長さボタンと合わせて使います。たとえば8分音符ボタンをクリックしてから付点音符ボタンをクリックすると付点8分音符を入力することができます。

連符ボタン

音符パレットの上の段にある付点ボタンの右の3という数字が書いてあるボタンは連符を入力するときに使うボタンで，このボタンの上でマウスの右ボタンをクリックすると3連符，5連符，7連符に切り替えることができます。このボタンも音符の長さボタンと合わせて使います。

たとえば8分音符ボタンをクリックしてから連符ボタンを3連符にしてクリックすると8分音符3連の音符の長さを入力することができます。

休符ボタン

音符パレットの真ん中の段には休符ボタンが左から順に「全休符」,「2分休符」,「4分休符」,「8分休符」,「16分休符」,「32分休符」と左から順に長い休符から短い休符になっています。音符を入力するときは休符の部分は開けて入力すれば休符は勝手に入るのでこのボタンはあまり使わないでしょう。

音符の強さボタン

音符パレットの一番下の段にあるのは音の強さボタンです。このボタンは音の強弱記号で音の強さを表しています。左から順に「ピアニッシシモ」,「ピアニッシモ」,「ピアノ」,「メゾピアノ」,「メゾフォルテ」,「フォルテ」,「フォルティッシモ」,「フォルティッシシモ」と左から順に弱い音からだんだん強い音になっています。

ツールボタンとは？

音符パレットの右にあるのがツールボタン類です。音符の入力を助けたり入力をやりやすくするための便利な機能があります。





このボタンは五線譜の画面の表示領域を拡大したり縮小するときに使います。拡大，縮小の幅は3段階あります。入力するときにしやすい大きさで入力しましょう。



このボタンは編集操作を元に戻したりもう一度やり直すときに使います。前に入力したデータとやり直したデータを比較するときに便利です。

その他のボタンについては入力のときに必要なところで説明します。

五線譜画面の説明

楽譜（スタッフ）画面の上半分の五線譜の画面の一番上に書いてある数字は小節番号です。番号の上にある青い線は小節の区切りで，その横のグレーの線はその楽譜で設定した拍子の基準となる音の長さ（例：4／4拍子の場合4分音符）を表しているので音符を入力するときの参考になります。

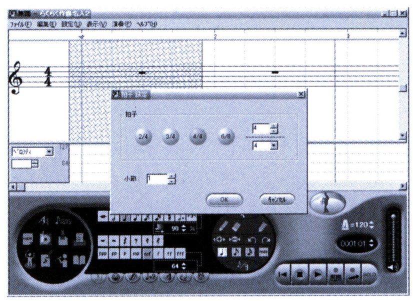
楽譜の設定

それでは音符入力に入りたいところですが，音符を入力する前に入力する曲の調と拍子を五線譜に設定しなければなりません。設定の方法は楽譜（スタッフ）画面の上半分の五線譜の画面の左端のト音記号等の音部記号のあたりをマウスで右クリックしてサブメニューを呼び出し，ダイアログから設定します。

また曲の途中でも五線譜の上の任意の場所でマウスの右ボタンをクリックするとサブメニューから調と拍子を変えることができます。

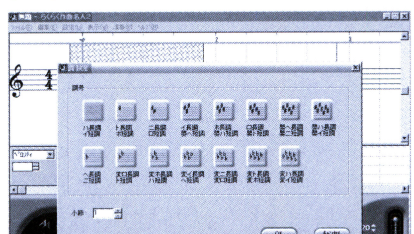
拍子の設定

サブメニューから拍子設定を選ぶと，拍子設定のダイアログが表示され拍子を設定することができます。主に使われる拍子はマルいボタンで用意されていますがその他の拍子は右の選択できる数字で設定できます。拍子を選んだら必ず［OK］ボタンをクリックして設定を有効にしてください。



調の設定

サブメニューから調設定を選ぶと調設定のダイアログが表示され調を設定することができます。入力したい調をクリックして選択します。必ず［OK］をクリックして設定を有効にしてください。



音符入力

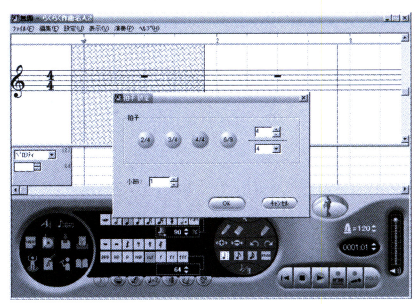
さあ，やっと音符を入力できます。普通DTMでは１小節目には演奏のための設定等が入るため，この『らくらく作曲名人』でも2小節目から音を入力するように1小節目は2重の斜の線で入力しないようになっていますね。だから2小節目から入力しましょう。

全音符や2分音符や4分音符を入力するときは，とくに注意はいりません。音符パレットから音符をクリックして選択してマウスのポインタが鉛筆ツールになったら五線譜に，音の高さと，どの拍に置くかをきめてクリックすれば入力できます。間違えた場合は，音符入力パレットの横のツールボタンの中から消しゴムのボタンをクリックして，マウスのポインタを消しゴムツールにして，消したい音符をクリックすれば消すことができます。

またツールボタンで，矢印や鉛筆ボタンを選んで間違えた音符上の左側にマウスポインタを置くと，マウスポインタが［＋］に変わるので，この状態でドラッグすると音符を左右に移動することができます。また，マウスポインタを音符上の右側に置くと，マウスポインタが上下の矢印に変わるので，この状態でドラッグすると，音符の音程を変えることができます。

8分音符等短い音符の入力

8分音符や16分音符や32分音符等の短い音符を入力するときは，音符パレットの右にあるツールボタンのグリッドボタンで短かい音符を入力できるようにしなければなりません。グリッドボタンは4分音符と8分音符と16分音符のボタンがあります。グリッドとは，設定した音符の位置で音を入力するための五線譜画面にある縦線のことです。たとえば，これが4分音符のまま1拍に16分音符を四つ置こうとしても置くことができません。そこで，グリッドをツールボタンから16分音符を選択してやると，16分音符を1拍に四つ置くことができます。ツールボタンで［free］ボタンを選択すると，グリッドが関係なくなるので，いちいちグリッドを変えなくて済むのですが，こうすると音が微妙にずれてしまうので，通常は16分音符にしておいたほうがほとんどの音符が入力できるのでよいでしょう。



臨時記号の入力

音符に臨時記号を付ける場合は，音符を入力するときに音符パレットから音符の長さボタンを選んだ後に，臨時記号ボタンをクリックして臨時記号を入力できる状態にして，五線譜に入力

します。入力した後に臨時記号ボタンをもう一度クリックして臨時記号の入力の解除を忘れないようにしましょう。

付点音符の入力

付点音符を入力する場合は，音符パレットから入力したい付点音符の付点を抜いた長さの音符の長さボタンをクリックして選んだ後に付点ボタンをクリックして入力します。付点音符を入力した後は，もう一度付点ボタンをクリックして付点音符の入力を解除するのを忘れないようにしましょう。

また，ツールボタンの中の上から2段目の右端にあるボンドツールを選択し二つの音符（同じ音程）をクリックすると，二つの音符をつなげて一つの音符にすることができます。たとえば，4分音符と8分音符をボンドツールでつなげると，付点4分音符が入力できます。

連符の入力

連符を入力するときは少し注意が必要です。音符パレットから入力したい連符の音符の長さを音符の長さパレットから選択した後に，連符ボタンをクリックして入力する前に，音符パレットの右横にあるツールボタンの中からグリッド入力を解除する［free］ボタンをクリックして，グリッド入力を解除します。そうしないと入力した音符がグリッドにそって入力されるため，連符の入力が正しく行われません。

また連符の入力が終わったら，ツールボタンから［free］ボタンをクリックしてグリッド入力を有効にし，音符パレットの連符ボタンもクリックして連符の入力も解除するのを忘れないようにしましょう。

休符の入力

休符は音符パレットの休符ボタンから入力できますが，普通は休符の部分はあけて音符を入力するので勝手に休符が入ります。休符ボタンで休符を入力する場合は，連続して音符を入力してあるところに休符を入れたいときに，休符ボタンから休符の長さを選んで五線譜の休符を入れたい位置をクリックして入力します。そのとき，休符を入力した位置より後の音符は，入力した休符の長さの分だけ後ろに移動します。

音に強弱をつける

入力する音に強弱を付けたい場合は音符パレットの音符の強さボタンを使います。入力の方法は，音符パレットから入力したい音符の長さを音符の長さボタンから選んだ後に，入力したい音符の強さを音符の強さボタンから選んで五線譜に入力します。とくに音符の強さを設定したくないときは［mf］（メゾフォルテ）か［f］（フォルテ）ぐらいにしておきましょう。

別に決まってないですが，後で部分的に強くしたり弱くしたいときに真ん中あたりの強さで入力していたほうが変更しやすいですから。

合わせて使う

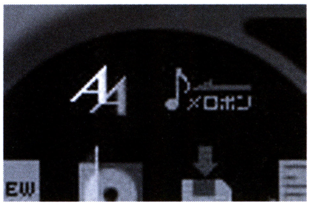
これまで説明してきた音符の入力はもちろん合わせて使うこともできます。たとえば［4分音符］ボタンを選択して［臨時記号］ボタンを［♯］にしてクリックして臨時記号♯を付けて［付点］ボタンをクリックして付点4分音符にし，音の強さボタンから［f］（フォルテ）をクリックして選択して五線譜にドの音を入力すると，ドのシャープの高さの音で長さは付点4分音符で音の強さはフォルテの音が入力されます。

他にもたくさんの組み合わせの入力がありますが，基本はこれまで説明してきたことなので，それを組み合わせて使ってもらえばこれだけでだいたいの曲が打ち込めるようになると思います。少しずつ音符の入力にも慣れましょう。

「らくらく作曲名人2」の楽しい使いかた

音符の入力ができるようになると，知っている曲のメロディを入力してスタイルを選択するだけで簡単にアレンジができます。最近は音楽雑誌にカラオケ用の楽譜が付いていたりしてますよね。それを携帯電話のチャクメロに入力する人もいるんじゃでしょうか？ それと同じ要領でらくらく作曲名人2に入力してしまいましょう。ステージ画面で足元の番号が1のプレーヤにメロディを入力します。その後はめちゃらくのデータをEDITしたときと同様にスタイルを選択して，オートアレンジボタン(自動的にアレンジしてくれるボタン)を押すだけです。すると，曲調を決めるダイアログが出てくるので，8種類の中から一つを選択して［OK］を押すだけです。

ちなみに，オートアレンジボタンはメロポンボタンの隣にあります。知っている曲をいろいろアレンジするとけっこうおもしろいですよ。



◆Webの内容

- 芸大生のらくらく作品
- EDITデータ見本・音符入力データ見本
- コンテストのくわしい応募方法

◆次回予告

- 「らくらく作曲名人2」アレンジ編①
- 東南アジアの音楽（コラム）
- シリーズ音響学（コラム）

「らくらく作曲名人2」作品募集！

前回までWeb上で行われていた「めちゃらく作曲名人」作品募集の中から「めちゃらく作曲名人大賞」，他いくつかの賞の発表がWeb上で発表されましたが，みなさんもう見てくれましたか？

今回からは，「らくらく作曲名人2」の連載が始まるということで「らくらく作曲名人2」の作品発表会を弊社ホームページ上で開催します。今回の募集作品は，前回まで使っていた「めちゃらく作曲名人」で作曲したデータを「らくらく作曲名人2」でEDITしたデータを募集します。作品の発表会は来月のWeb上で開催しますので，どしどしご応募お待ちしています。参加希望の方は以下のデータをご用意のうえ，「らくらく作曲名人2・作品募集」係まで郵送かまたはEメールで（アドレス）に添付してお送りください。

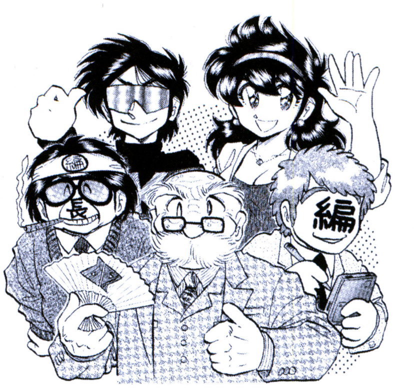
音色やテンポを変えるだけの簡単なEDITでもかまいませんよ！

・ご用意いただくもの

①EDIT前のめちゃらく作品データ，MIDI･xwsファイル
②EDIT後のらくらく作品データ，MIDI･xwsファイル
③作品の簡単な紹介，コメント
④作曲者プロフィール
⑤連絡先（Eメール・アドレス）
（郵送の場合は，フロッピーへ）

http://www.beeplus.or.jp/bm/art/

あくせすROOM



OOF
ダブルオーエフ

♥激しい腹痛で学校を早退し，内科で点滴してもらいました。翌日は休み，翌々日に登校すると「ガンで夏休みに手術」「今は薬でだましだまし」とゆーことになってました。まだ死にゃせんよ！

（東京都練馬区・ルストフレム 14歳）

つぐ美：本当に危なかったら，夏休みを待たないでしょ。大丈夫，大丈夫！

影：病気にでもなって休みたいねえ。

Dr.D：このバチ当たりが！

編：たまには，病気になるぐらいの仕事をしてください。

♥つぎからつぎへとアイデアをひねり出すベーマガ読者には頭がさがります。

（宮崎市・ktr 16歳）

Dr.D：若いモンがなに言っておる。

編：授業中とかって，どうでもいいことがいっぱい思いつきますよね。そういうのを専用ノートにメモしていると，1年でものすごい量になった気がします。

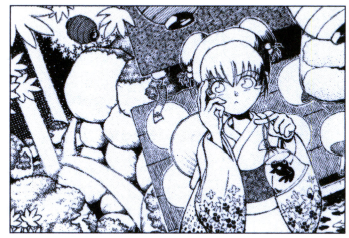
影：自分ではくだらないなあと思っても，見る人が見たら，アイデアの宝庫だったりするんだよな。ネタノートの交換会なんてなかった？

♥絵描いといて……なんだけど，主役はプログラムのページなんだから，お絵かき小僧は目立たなくても良いですよ……。そして，そのぶんをプログラム・リストが埋めるのだ……!! PS2の，早く載らないかな。

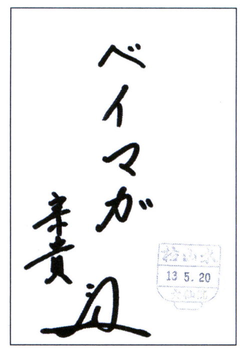
（愛知県名古屋市・わいち301 27歳）

影：主役だけじゃあドラマは作れないぜ！ 編集部だって，私の引き立て役がいっぱいいるからこそ……。

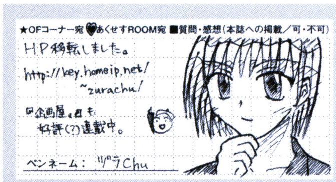
編集長：そんなに主役を張りたいなら，ビチアス海淵支局にでも飛ばして，1人舞台を踏ませてやってもいいぞ。んん？



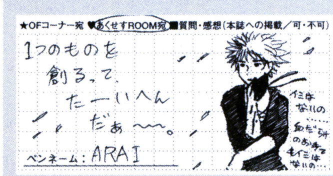
◀愛知県名古屋市・わいち301
つぐ美：涼しげな夏祭りのヒトコマ。影：今年は暑い！ とにかく熱い！



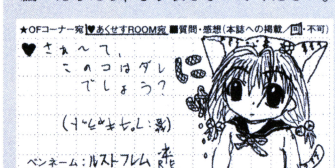
◀東京都練馬区・ルストフレム
編：「大仙院のおしょうさんに書いてもらいました係御中」って……？



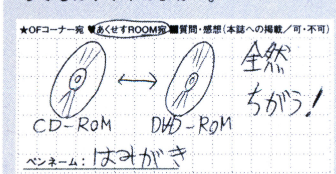
▲和歌山県橋本市・ヅラchu
編：バリバリがんばってください。
影：このスペース、宣伝に最適だな。



▲大阪府吹田市・ARAI
影：そう、とっても大変なのだ！
編：だったら、もう壊さないでください。



▲東京都練馬区・ルストフレム
つぐ美：帽子の表情が変るってのが、とってもカワイイのよね〜。



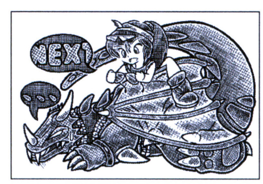
▲北海道天塩町・はみがき
影：ぜんぜんわからん！

素材名人 萌えプロ

影：ちょっと感動した素材を組み合わせて，大きな素材に組み立てる「萌えプロ」。テーマものだけに掲載チャンスはその月一度かぎり。なるべく没を減らしているつもりなのだが……どういうわけか，ほかのイラスト葉書よりも投稿が多すぎるのだ！ あな恐ろしきは常連なり。



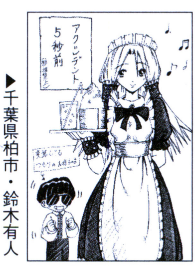
◀神奈川県大和市・TNT爆弾
編：大きめの紙に出力してから、余白を切断すれば、タチキリがきれいですよ。



◀愛知県刈谷市・せゆ
影：陰影にもっとメリハリをつければ。フリルも目立ったんだけどなあ。



◀福島市・南斗ライバ
つぐ美：サングラスをかけて日焼けすると、本当にパンダになっちゃう〜。



◀千葉県柏市・鈴木有人
編：自分の視界が変わると、相手からもちがって見えていると錯覚しますよね。

お題と締切日

影：参加しやすいようにテーマ数を減らしてみたぞ。

9月号	10月号	11月号	12月号
・お供（ペットや従者）	・妹	・メイドさん	・防災訓練
・サングラス	・せまいところが好き	・楽器が好き	・菓子パン
・フリルいっぱい（服に限定せず）	・西部開拓時代	・（自由課題）	
（締切：2001年6月30日に終了済）	（締切：2001年7月30日）	（締切：2001年8月30日）	（締切：2001年9月30日）

あくせすROOM応募のルール

●愛読者カードや，官製ハガキ，同サイズの丈夫な紙などでお送りください。

ワンポイント
まとめ送りで大量に送るよりも，気合いの入った数枚の作品のほうが，インパクトは強く，掲載率は上がるようです。

●鉛筆や色鉛筆，涙滴の多いボールペン画は印刷に不向きなので避けましょう。

ワンポイント
下書きのあとが消し切れてないと見苦しくなり，評価が大幅にダウンしてしまいます。仕上げはていねいに。

●投稿内容は，掲載されたプログラムの感想や，学校でのできごと，一発ギャグなど，なんでも結構です。みなさんで楽しいページにしていきましょう。

≪応募先≫
〒141-8715 東京都品川区東五反田1-11-15
電波新聞社マイコンBASICマガジン編集部
あくせすROOM係

webでも受付中です。Internet読者ハガキでどうぞ。
http://www.basicmagazine.net/

大人気クリエイター集団「ギガ連射」があれこれ語る

GIGA RENSYA つつうらら

•第12回•

グラフィッカーのラナスです。みなさん夏バテせずにがんばっていますでしょうか？めちゃめちゃ暑い日々をお過ごしかと思われます。これを書いている時点で北海道はまださほど暑くもなく，夜になると外はまだ薄ら寒い感じがします。さて今月のつつうららは？

『ちゃくちゃくと進むクター』

ギガ連射のHP自体はとくに進展もなく，なにもしていないように見えますが，じつはちゃくちゃくと進行しています。かなり手の込んだゲームも作っています。まだ公にはできませんが，この号が出る頃にはてんこ盛りになっているかもしれません。

アスレチックのニューゲームも出てるかな？

『教えてギガ連射ー拡張版』

“ベーマガ”で1年間も連載させてもらっていると，結構お便りメールがくるんだな。そこで，今回はラナスが独断と偏見でお答えしちゃうぞ。よりギガ連射を身近に感じてくれぃ！

★ ★

Q：どうして，ギガ連射という名前なんですか？？？

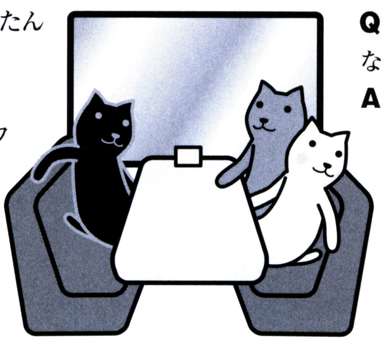
A：これ，よく来る質問だね。マウスをめちゃめちゃ連射するからだ。紆余曲折の末，”ギガ連射”に落ちついた…なんだかなぁ。

Q：誰がクターを考え出したんですか？

A：オレだ。

Q：クターのタイピングソフトをだしてください。

A：マウスでできないだろう！まぁそのうち考えてみようかな。こういう要望がとても多いんだな。



おもしろげなネタは随時募集中！

Q：みなさんの趣味は何ですか？

A：オレはサッカー観戦が趣味だ。この前も札幌ドームで日本代表戦を見てきたぞ。あとはみんな大体ゲーム好き。

Q：みなさんは彼女とかいますか？

A：オレは1人ゲットした。2人目は修羅場になりそうなのでやめておく。他のメンバーは別れや再会などいろいろあるようだ。

Q：ギガ連射のみなさんはブリーフ派ですか？トランクス派ですか？なんとなく危険なブリーフな香りが漂ってきます。

A：少なくともオレはトランクス派だが全裸派でもある。やっぱり股間は自由なほうが良いだろう。他のメンバーはブリーフが似合いそう。

Q：ギガ連射の方々が好きなゲームは何ですか？クター以外でです。ダメなら好きなゲーム機は何か教えてください。お願いします。僕はニューファミコンが好きです。

A：オレはパソコン用のディアブロ2の拡張パックがマイブームだ。仕事する時間を忘れるぞ。好きなゲーム機はドリームキャストだ。メンバーはわりとセガ党が多いぞ。

Q：ところで，初恋はいつなんですか？

A：小学校1年生のときのとなりに座ってた子だ。オレが転校したのでそれっきりだ。

Q：リョウコ（ギガ連射サイトで壁紙のみ公開したキャラクター）とやらは，どこに行ったんですか？聞いちゃいけないことなんですか？気になって，夜と授業中しか眠れません。ってゆうか，リョウコって，何者ですか？

A：リョウコは新手のネットアイドルになる予定だった。誰かさんの好みのタイプだったのかもしれない。聞いちゃいけないことではないが，そっとしておいてほしいみたいだ。夜は寝てもいいが，授業中はしっかりと話を聞くように。

Q：ギガ連射のメンバーのみなさんが一番好きな曲はなんですか？

A：オレはJ-POP派だけど，カラオケではキーがあわずに歌いきれないので，アニメソングがちょうどよい。

Q：みなさんはギガ連射での収入のみで生活しているのでしょうか？ギガ連射の収入源はどういったものでしょうか？

A：ギガ連射の収入源はクターを使った他社とのお仕事だ。今はまだ目立たないが，今にドーンとでかいことしでかしそうな予感。みんな，待ってろ！

★ ★

どうだろう，少しはギガ連射を理解してくれたかな？アニメソングしかうまく歌えなくても，出会いや別れに枕をぐっしょり濡らしても，はたまたブリーフ派だと疑われても，おもしろいゲームってのはできるモンなんだ。ゲーム作りたい人は，これを参考に前向きに生きろ！

感想・質問・交際の申し込みなど，みなさんからのメールお待ちしています。
otayori@giga-rensya.com
ギガ連射WEBサイト
http://www.giga-rensya.com/

SEGAのヘンな話題やちょっとお得な情報ならおまかせ!!

ベーセガ

マイコン BASIC
SEGA OFFICIAL VARIETY SPACE

竹崎 奈美〜，オー・マイ・奈美！ 大変だよー。

奈美 なんですか，竹崎さん。もう「ベーセガ」の本番始まってるんですから，ふざけてちゃダメですよ。

竹崎 ちがーう！ マジマジマジマジ超マジよ！

奈美 あら，そうだったんですか？

竹崎 なんと，聞いて驚け！ 急遽，我々のオフィスが羽田に戻ることが決定したのだっ！

奈美 えっっ（絶句）！ 1年前の悪夢再び！ 休日出勤＆毎晩深夜までの荷造り作業…。そ，そんなぁーー。

竹崎 おいおい，それはちとオーバーじゃない？

奈美 …さようなら竹崎さん，長い間お世話になりました（ぺこり）。

竹崎 おーい。奈美さん，戻っておいで〜！ 以上 “おたよりの宛先変更に関するお知らせ”でした〜。ところで奈美さん，僕らもこうして長くコンビを組んでやってきたわけだが，ここらでひとつ慰安の意味を込めて，温泉にでも行かないかね？

奈美 「温泉」ですか？ いいですねえ！

竹崎 でしょ！ 伊豆かな…熱海かな……。

奈美 はい！ これのことですよね，どーーん!!「ぐるぐる温泉2」!!

竹崎 は？

奈美 何って，8月9日発売のDC用ソフトですよ！

竹崎 あの……伊豆は熱海は露天風呂は……。

奈美 ハイ「ぐるぐる温泉2」は，チャットにテーブルゲームに，と一躍人気者となったネットワークゲーム「ぐるぐる温泉」の続編なんですねえ。

竹崎 ネットワークゲームで，なんと言っても楽しいのは，いろいろな人達とのチャットだよね！

奈美 「ぐるぐる温泉2」では，このチャット機能を大幅に強化。

竹崎 チャットルーム（部屋）も用途に合わせて，複数タイプをご用意しました（笑）。

奈美 自宅に居ながらにして旅行気分を味わえる，「特急列車『おんせん』」ですとか，夏の夜の怪談話に最適（？）な「夜の廃屋」，簡単なクイズを楽しめる「温泉クイズショー」などなど。見た目も種類も豊富ですっ。

竹崎 ただ文字によるチャットを楽しむだけでなく，プレイヤーキャラクターには，笑ったり怒ったりと，さまざまなリアクションをとらせることもできるぞ。

奈美 リアクションによって，フキダシの形や色も変化しますから，ビジュアル的にも楽しめますね！

竹崎 プレイヤーの分身となるキャラクターは，パーツ数を前作よりも大量に増加！

奈美 さらに色指定も可能になりましたし，目と目の間の位置調整なんかもできるようになりましたから，より個性豊かなキャラクター作りができます。

竹崎 なるほど。これで見知らぬ誰かとキャラがかぶることもなくなるわけか。

奈美 お馴染みの名刺機能もパワーUPしてます！

竹崎 名刺にはプロフィールやコメントを記入できるだけでなく，名刺交換をした相手を検索することもできるので，直接会いに行ったり，簡易メールである「電報」を送ったりなどもできます。相手がオフラインの状態であっても「伝言」メールを送ることができるので，ばりばり活用しよう。

奈美 これは便利な機能ですね〜。

竹崎 名刺はスケジュール帳形式で管理できるし，前作で集めた名刺もコンバート可能だから「ぐるぐる温泉1」で知り合った友達にもすぐに会えるぜっ。

奈美 名刺を使いこなして，目指せ温泉マスター！（笑）

竹崎 その他「文字放送機能」なんてのもあるぞ。

奈美 これは，オンライン接続中に文字放送が画面上部に流れる，という機能ですー。

竹崎 チャンネルも複数用意されているから，自分の知りたい情報をリアルタイムに入手できるのだー。

奈美 さて「温泉2」に収録されている肝心のゲームのほうですが，老若男女を問わずに大人気の「UNO」。数あるトランプゲームの中でも高度な駆け引きが楽しめる「ポーカー」。

竹崎 前作から引き続き登場となる「麻雀」。そして，初のオリジナルゲーム「温泉捜査課」の四つが用意されているぞ。

奈美 「温泉捜査課」は，プレイヤー同士が協力して犯人を追う，というゲームなんですよ。面白そうでしょ？

竹崎 ナルホド。まさに至れり尽くせりという訳だ。仕方がないから本物の温泉はあきらめるかー。「温泉2」よろしくねッ！

奈美 さーて，本番も終わったことだし，私は本命と本物の温泉，行ってこようかなー。

竹崎 ずがーーーん！（笑）



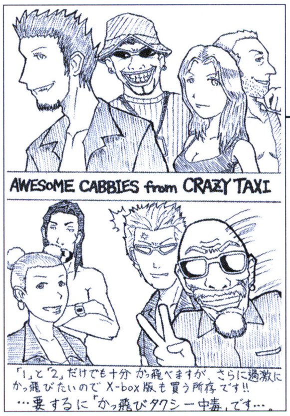
◀葉月魁斗

あれは私，ゲーマー奈美なんですヨ。期待させてしまってごめんなさい〜



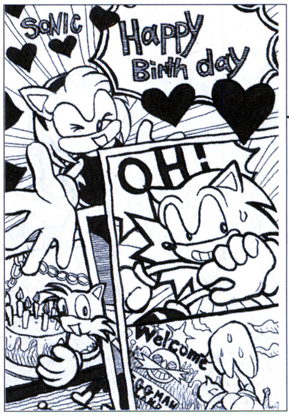
◀ARAI

シャドウだそうです。誰が何と言ってもシャドウなのだそうです…



◀バグニンゲン

おハガキありがとうございます。亀○人ですか，ナルホド…



◀ぱんだぱぱ

むぎゅー。雰囲気がとてもよく出てますねえ。アメリカの風を感じます（笑）

このコーナーを充実させるための，ご意見，お便り，イラストなどなどクビを長くしてお待ちしています。採用の方には特製テレカなど，非売品セガグッズをプレゼント。SEGA竹崎がお便りをすべて読んで，面白いコーナーを作っていきたいと思います。すべてのあてさきは，ここや!!

〒144-8531 東京都大田区羽田1-2-12 （株）セガ カスタマーゲートウェイ部「竹崎と奈美のベーセガ」係

セガ竹崎の「セガスタ」はこちら→http://www.sega.co.jp/segasuta/

Falcom® NEWS EXPRESS

日本ファルコム株式会社
〒190-0012 東京都立川市曙町1-14-13
TEL 042-527-6501
FAX 042-528-2714

スタッフ募集中
採用担当BFN係
http://www.falcom.co.jp/

ファルコムサウンド大特集！

今月は，ゲームミュージック史を語る上で欠かせないファルコムサウンドの大特集！1988年に音楽業界でゲームミュージックの地位を大きく向上させる原動力となったファルコムレーベルを発足。これまでに発売されたタイトルは実に140枚以上！サントラ盤をはじめとして，シンフォニーやロック，ピアノ，ボーカルなど様々なジャンルにアレンジされ，また羽田健太郎，つのだ☆ひろ，井上大輔，難波弘之，北島健二，鳴瀬喜博，米光亮などもはやゲームミュージックの域を越えた超豪華なミュージシャンも参加。そして，ユーザーの方のリクエストをもとにファルコムミレニアムシリーズ（FMS）として，ファルコムレーベルからの復刻盤と完全新譜が発売，現在23タイトルが発売中ですが，いよいよ第4弾の発売が決定！ファルコム作品の中でも特に人気の高いガガーブトリロジーとイースⅠ・Ⅱから3タイトルです。ご期待ください。

FMS第4弾，近日発売決定！

1 交響曲ガガーブトリロジー～白き魔女～朱紅い雫～海の檻歌（新譜）

3部構成からなる圧倒的密度（収録時間70分！）の超大作交響曲。完全新作新アレンジで堂々完成！
【収録曲】第1部「朱紅い雫」伝承／絆／朱紅い雫　他，第2部「海の檻歌」Leone Fredrik Richter “海の檻歌”／キャプテン・トーマス！／水底のメロディー　他，第3部「白き魔女」小さな英雄／それ見よ我が元気／白き魔女　他

2 パーフェクトコレクションイースⅠ・Ⅱ～米光亮全曲集（2枚組）

イースアレンジ決定版！パーフェクトコレクションイース＆イースⅡ米光亮全曲アレンジ完全復刻版！
【収録曲】FEENA／FIRST STEP TOWARDS WARS／PALACE OF DESTRUCTION／THE MORNING GROW／TO MAKE THE END OF BATTLE／LILIA／ICE RIDGE OF NOLTIA／PALACE OF SALMON／SUBTERRANEAN CANAL／STAY WITH ME FOREVER 他

3 パーフェクトコレクションイースⅠ・Ⅱ～スーパーアレンジ全曲集（2枚組）

OVA「イース」エンディング曲「Endless History／南翔子」収録！名盤パーフェクトコレクションイース＆イースⅡ完全復刻版！
【収録曲】ボーカルVer.:Endless History（THE MORNING GROW）／Smile Again（SEE YOU AGAIN）他　林有三アレンジVer.:FEENA FIRST STEP TOWARDS WARS　他ニューエイジVer.：PALACE／REST IN PEACE 他，J.D.K.アレンジVer.：TOO FULL WITH LOVE 他

第1弾～第3弾 好評発売中！

イースシリーズ

■交響曲イース
■フィーナ／南翔子
■イースピアノコレクション
■オリジナルサントライース&イースⅡエターナル
■オリジナルサントラ「イース天空の神殿」J.D.K.バンド編
■オリジナルサントラ「イース天空の神殿」寺嶋民哉eオーケストラ編

英雄伝説シリーズ

■オリジナルサントラ海の檻歌　前編
■オリジナルサントラ海の檻歌　後編
■オリジナルサントラ新白き魔女　前編
■オリジナルサントラ新白き魔女　後編
■オリジナルサントラ朱紅い雫
■交響詩英雄伝説Ⅲ
■英雄伝説Ⅰ～Ⅳピアノコレクション

ソーサリアンシリーズ

■ソーサリアン・スーパーアレンジバージョン
■ソーサリアン・スーパーアレンジバージョンⅡ
■ソーサリアン・スーパーアレンジバージョンⅢ
■交響曲ソーサリアン

その他

■ファルコムスペシャルBOX'89①
■ファルコムスペシャルBOX'89②
■プリプレマ
■ザ・ボーカルフロム・イース・サラウンドシアターソーサリアン&プラスミックス
■ファルコムJ.D.K.バンド1
■イース&イースⅡ，朱紅い雫 Unpublished Music

ファルコムのCDはこのお店でGet！

通信販売以外でも，FMSシリーズは下記販売店で好評発売中。お近くの方はぜひご利用ください。

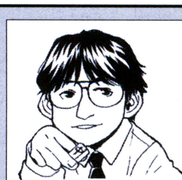
●メッセサンオー本店A館2F
東京都千代田区外神田3-14-8
03-3255-3452

●ヤマギワソフト館
東京都千代田区外神田3-13-12
03-3253-5111

●ヤマギワソフト大阪店
大阪市浪速区日本橋3-4-9
06-6643-3331

●ゲーマーズ全店
本店（03-5298-4588）
新宿店（03-3225-5996）
スクエア店（03-3252-9600）
立川店（042-548-8577）
長野店（026-266-7626）
なんば店（06-6630-0404）
博多店（092-434-6868）
池袋店（03-3985-3232）
秋葉原店（03-5298-2015）
水道橋店（03-3296-8337）
横浜店（045-317-4321）
名古屋店（052-957-3303）
京都店（075-255-1700）

ここに紹介している販売店はごく一部。この他にもファルコムホームページで，取扱店を全て掲載しているので要チェック！
Http://www.falcom.co.jp./Shoplist/



FMSシリーズもいよいよ第4弾！今回発売されるパーフェクトコレクションイースⅢは，発売当時オリコンでも上位にランキングされた，シリーズを通して最も人気の高かったタイトルです。いろんなアレンジが楽しめますので，ぜひ聴いてみてくださいね。



タイトル冠文：御堂筋太郎
タイトル・イラスト：大阪府・藤井朱丸
本文：暁

ブロードバンド対応 なにわ通信

PRESENTED BY マイコンソフト ADDRESS● 〒530-0005大阪市北区中之島3-2-4 朝日新聞ビル6F 電波新聞社大阪本社内 マイコンソフト株式会社 なにわ通信係
http://www.micomsoft.co.jp/

みなさんこんにちは。

学生さんは夏休みですね。皆さんは何処か旅行に行かれたのでしょうか？

私はユニバーサル・スタジオ・ジャパンに行ってきました。映画が好きな人にはうれしいアトラクションがいっぱいです。

個人的にはバック・トゥーザ・フューチャーのアトラクションがお奨めです。

新型アップ・コンバーター

電波新聞社より発売されております，アップ・スキャンコンバーター・ユニットXRGB-2の後継機を開発していることは，以前お伝えしましたが，この度，その後継機の写真を入手しましたので，ご紹介させていただきます。

ご覧のとおり，新たに本体前面にD1入力端子が付きましたので，D1出力端子付きDVDプレイヤーや今後発売されるゲーム機の鮮明な映像を入力できます。今回は，XRGB-2よりパワーアップした機能をご紹介します。



パワーアップした機能

- ●S入力の鮮明度が更に良くなった。
- ●ガンマ調整が96段階できるようになった。
- ●D1入力端子が追加された。
- ●21ピンRGBマルチ端子からの色差信号入力で画面が緑色になってしまう問題を改善した。(色差信号/RGB信号の手動切替対応)
- ●水平同期周波数31kHzに加え水平同期周波数47kHzの画面モードにも対応できるようになった。

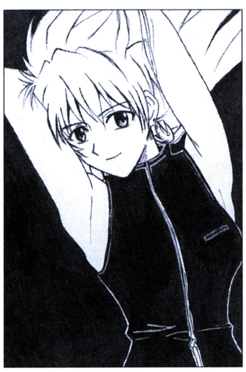
パソコン用ディスプレイでゲーム画面やDVD映像を見たい方にはオススメの装置です。

発売は11月を予定しています。

ありがたいおハガキのコーナー



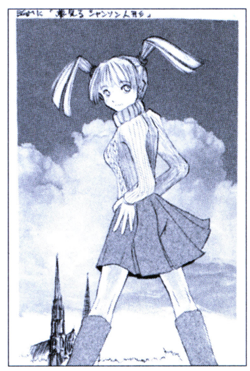
三重県・猿ともこ
焦らず，自分の良いところを探して，自分を好きになれるよう努力してみてください。あなたにも良いところが沢山あるはずです。



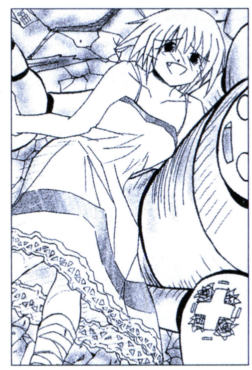
熊本県・特異点
セクシーなポーズですねぇ〜。黒い服がセクシー度をアップさせているように思います。 また投稿してくださいね。



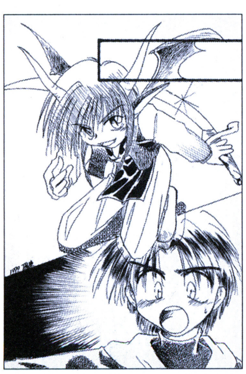
千葉県・鈴木有人
どこか違う世界の，大きな街の昼下がり。その街には異国の商人が集まる。私は獣の顔を持つ武器屋の前で足を止めた。



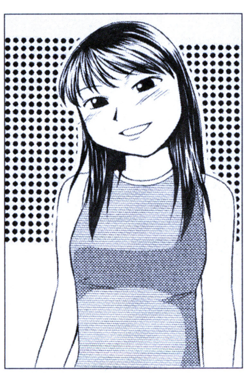
大阪府・高原久明
空が大きく開放的なので大好きです。私は，下から上を眺める感じの絵が好きみたいです。



大阪府・地震大変でしたね。
小説のワンシーンでしょうか？どんな小説なのでしょうか。読んでみたいので作者と題名をぜひ教えてください。



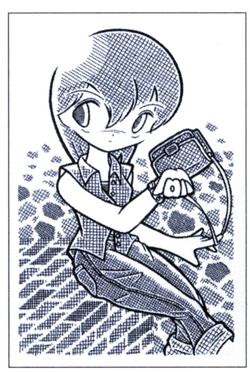
京都府・陽東柾樹
陽東さんはNHKマニアです。NHKアーカイブズで放映していた「夢の島少女」というドラマがすごく良かったですよ。



埼玉県・ぶんちょ
こっ，これは水着ですよね！肩の辺りが，かなりセクシーです！30才過ぎると視点がオヤジっぽくなりますね(笑)。



神奈川県・Lフィ
はじめまして！そして掲載が遅れて申し訳ありませんでした。丁寧に描かれている，とても素敵なイラストです。



愛知県・せゆ
速度を必要としないプログラムならVBでもかまわないのです。しかし，プログラマで食って行くにはC言語の勉強は必須ですよ。



宮城県・戸田ようこ
猫はいじめない方がいいと思いますよ(笑)。私は昔，猫にインスタント麺の「どん兵衛」を横取りされたことがあります。

ハムフェア2001 パシフィコ横浜で開催!!

8月31日(金)/9月1日(土)/9月2日(日)

来て!見て!触れて!JAIAコーナーで各社の最新情報をゲットしよう!!

JAIAコーナー出展各社
- ●最新機種を見て!触れて!楽しめる!
- ●アマチュア無線の最先端をじっくりご覧ください。

●アイコム株式会社　●第一電波工業株式会社
●株式会社ケンウッド　●日本無線株式会社

人と人を電波で結ぶアマチュア無線。あなたも、始めてみませんか?

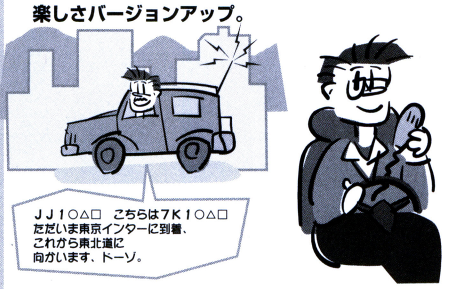
アマチュア無線は、電波を使って仲間同士はもちろん、世界中のアマチュア無線愛好家とコミュニケーション。
年齢や性別にとらわれることなく、誰にでも楽しめる素晴らしい趣味であり、社会奉仕にも大いに役立っています。

こんな所で。　こんな時…

オンロード

クルマに無線機を搭載して楽しさバージョンアップ。

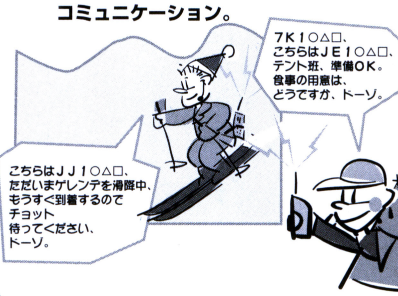
ドライブしながらハム仲間と交信。クルマからクルマへアマチュア無線でコミュニケーション。



アウトドア

野山から、手軽に電波でコミュニケーション。

スキー、登山、ハンググライダー、パラグライダーなどのスポーツ。ハイキング、キャンプなど、携帯用のトランシーバーがあれば楽しさアップ。



マルチメディア

コンピュータとアマチュア無線がドッキング。

最新のアマチュア無線は、声とモールス符号だけではなく、アマチュア無線の電波で文字や画像も送受信できます。



ボランティア

非常通信は、地震や大きな災害時に威力を発揮。

全国に非常通信ネットワークを構築しているアマチュア無線は、社会に貢献できる趣味として注目を集めています。



ホーム

サークル、自宅、学校。身近な仲間から遠く海外まで、広がるハムコミュニケーション。

自宅やクラブ局からのハムライフ、日本中のハムや遠く海外のハムとDX(遠距離)交信。



さあ免許を取ろう!

アマチュア無線の免許は、まず初級の第4級から取るのが近道。
この資格を取得するだけで、広範囲の楽しさを味わうことができます。

取得方法
養成課程講習会を受講(2日間)して取る

詳しくはJAIAへお問い合わせ下さい!!

JAIA 日本アマチュア無線機器工業会
〒170-0002 東京都豊島区巣鴨1-10-5　TEL 03-3944-8611

JAIA:日本を代表するアマチュア無線機器、アンテナ、周辺機器メーカーによって組織されている団体です。
URL http://www.jaia.or.jp

「JARD養成課程講習会」では、国家試験免除で、第4級アマチュア無線技士の免許が取得できます。受講時間はたったの10時間。平日、夜間、日曜祝日、夏休み・冬休みなど、あなたの時間に応じてお選びいただけます。JAIA専門会員は「JARD養成課程講習会」の事務の一部代行を目的として全国的に組織されています。受講の申込みから免許取得後の開局まで、あなたのHAMライフのすべてをお手伝いいたします。

JAIA専門会員店はベストアドバイザーです。

養成課程講習会

高い合格率！▶約98%

最短▶2日間/10時間

JAIAのお店でまってます。

ホームページもみてください。
http://www.jaia.or.jp

㈱富士自動車電機	熱海市桜町17-46	0557-83-2111
あしざわ無線	富士宮市宮原453-6	0544-24-1064
㈱ビーム	焼津市三ケ名1341-1	054-629-1513
富士・システム通信㈱	掛川市南2-1-19	0537-24-5350
むせんＺＯＮＥ２５	田方郡修善寺町日向362-4	0558-72-2961
岐阜県		
㈲サカイ無線	岐阜市福住町1-3	0582-51-5899
ＣＱオーム	岐阜市菅生3-11-8	0582-94-3949
㈲ハムショップヒダ電子	高山市下岡本町2558-24	0577-32-5665
愛知県		
㈱アイチムセン	名古屋市北区西志賀町1-90	052-912-0913
平丸ムセン	名古屋市東区東大曽根町19-3	052-914-7542
㈲名南無線	名古屋市港区高木町2-25	052-382-1094
タケウチ電子㈲	豊橋市大橋通2-132-2	0532-52-2684
センガ無線	半田市柊町4-201-38	0569-22-5571
㈲フジ計器通信機	碧南市金山町5-88	0566-48-4949
㈲トヨタムセン	豊田市日之出町1-4-6	0565-34-1253
㈱ロッキー電子	安城市錦町2-3	0566-75-3736
㈲新平和無線	額田郡幸田町大字芦谷字仲田16-1	0564-62-2281
三重県		
㈱電化パーツ津店	津市八町2-1-4	059-229-2737
信和通信機販売三重	南牟婁郡鵜殿村2151	0735-32-0167
京都府		
京洛無線	京都市下京区大宮通松原角	075-811-8051
シグマ無線	京都市伏見区桃山与五郎町1-46	075-623-0550
日栄無線㈱本店	京都市右京区西院六反田町31-4	075-312-6145
富士電機峰山店	中郡峰山町新町1568-4	0772-62-5593
滋賀県		
㈱中川電機	長浜市三ツ矢町9-18	0749-62-1366
日栄無線㈱滋賀店	守山市吉身3-17-29	0775-83-5108
奈良県		
アクティブ通信	大和郡山市美濃庄町291-1	0743-55-5169
岩本電機	吉野郡大淀町北野83-13	07463-2-9555
㈱奈良エコー無線	生駒郡斑鳩町法隆寺東2-2-19	07457-4-1164
大阪府		
大阪無線㈱	大阪市浪速区日本橋5-3-4	06-6643-0404
中野無線㈱	大阪市浪速区日本橋4-9-16	06-6641-8466
㈱ニノミヤエレホビー店	大阪市浪速区日本橋5-6-19	06-6632-2038
日本橋ハム	大阪市浪速区日本橋東2-2-2	06-6633-2922
㈲みさと無線	大阪市浪速区日本橋西2-7-3	06-6644-0111
日栄ムセン堺店	泉大津市東豊中2-1-6オレンジビル2階	0725-43-1059
長野ラジオシャック	河内長野市小山田町945-1	0721-54-5649
和歌山県		
トーラス	和歌山市東紺屋町6	0734-32-8614
㈲マルヤスデンキ	和歌山市田中町4-85	0734-36-3651
岡本無線	田辺市今福町39	0739-24-7555
マルミヤムセン	橋本市学文路141-4	0736-33-1771
兵庫県		
㈱ウエダ無線神戸	神戸市中央区三の宮1-3-5	078-327-3350
㈱正電社	姫路市立町74	0792-89-0396
㈱有楽庁	姫路市四郷町見野478-1	0792-53-0739
阪神通信機㈱	尼崎市大物町2-7-3	06-6481-7781
東田テレビ ・	洲本市本町4-5-13	0799-22-1403
日栄ムセン氷上店	氷上郡氷上町横田136-4	0795-82-2939
富山県		
中部特機産業㈱富山店	富山市今泉55-3	0764-91-6386
ＴＭＴサービス富山店	富山市豊田町2-9-17	0764-42-2963
無線パーツ㈱根塚店	富山市根塚町423-9	0764-25-6822
無線パーツ㈱高岡店	高岡市永楽町2-4	0766-25-6822
㈲沢田通信	砺波市千代168	0763-33-3654
福井県		
中部特機産業㈱福井店	福井市御幸3-13-24	0776-26-1641
㈱マルツ電波本店	福井市豊島2-7-4	0776-22-0202
石川県		
中部特機産業㈱金沢店	金沢市新神田1-9-24	0762-91-0818
無線パーツ㈱金沢店	金沢市西念町113-2	0762-31-3214
㈲無線屋	金沢市八日市3-550	0762-69-0980
中部特機産業㈱小松店	小松市矢田野町ホ124	0761-43-2419
岡山県		
㈲青電舎西岡山店	岡山市白石西新町13-108	086-254-5000
ハムス岡山	岡山市田中103-105-102	086-242-0200
ウエスタンレディオ	倉敷市大内1195-2	086-421-0080
クラハム倉敷店	倉敷市美和1-4-23	086-425-1300
ＣＱ水島	倉敷市連島3-2-22	086-448-9548
㈲丸善電気	津山市林田81-5	0868-31-3739
島根県		
㈲デンケンパーツ	松江市学園南1-5-7	0852-22-0236
㈲ナオラ	出雲市今市町1602	0853-23-0151
柳井無線パーツ㈱	益田市下本郷町999-1	0856-23-4686
アートムセン	八束郡東出雲町出雲郷863-1	0852-52-5100
山口県		
春日電子	宇部市西梶返3-10-34	0836-34-2178
㈲志賀電子	宇部市野原1-5-9	0836-31-8664
ドイ音響無線	徳山市代々木通り2-14	0834-21-6820
電子センター	防府市国衙1-11-1	0835-38-4949
岩国ハムセンター	岩国市平田5-30-14	0827-31-4949
松本無線パーツ㈱	岩国市麻里布町4-14-24	0827-24-0081
鳥取県		
高橋無線	鳥取市相生町1-610-2	0857-22-5735
㈲ニシデン米子	米子市米原6-18-32	0859-32-3463
広島県		
ハムショップマッコイ	広島市西区東観音町8-21	082-295-0887
ハムセンター広島	広島市西区観音町10-3	082-294-0566
広島パーソナルセンター	広島市安佐南区八木5-4-5	082-873-6515
松本無線パーツ㈱広島店	広島市中区銀山町2-6	082-243-4451
三原ハムセンター	三原市宮浦町3-25-11	0848-64-5032
徳山電子パーツ㈱	福山市本庄町中2-6-22	0849-21-1045
㈲ミカド電業	福山市御門町1-1-16	0849-22-2606
第一受信サービス	三次市東酒屋町162-1	08246-3-8265
ハムショップ東広島	東広島市西条町田口2931-42	0824-25-5286
香川県		
㈱電化センター	高松市天神前4-35	087-862-6077
ミナミ無線機器	坂出市江尻町1772-9	0877-44-3354
豊芝電機㈱	三豊郡高瀬町新名774	0875-72-3151
徳島県		
㈱ハムトク	徳島市沖浜東3-25-1	0886-26-0860
㈲サンベスト	阿南市学原町前田16-3	0884-23-6297
愛媛県		
㈲永田無線	松山市築山町12-5	089-931-4949
㈱大倉通信機今治店	今治市北高下町1-1-17	0898-23-0825
田中無線	宇和島市築地町2-2-19	0895-22-7200
㈱伊予通信機	新居浜市西喜光地町8-30	0897-44-5995
高知県		
アロックス㈲	高知市城山町218-8	0888-33-0366
㈲高知オーディオ・ハムセンター	高知市若松町6-6	0888-83-5411
㈱高知通信機ハムショップ高知店	高知市鴨部1-12-17	0888-44-4222
㈲土佐第一電子	高知市小津町9-4	0888-25-0335
㈱高知通信機中村営業所	中村市五月町2-12	0880-34-6035
福岡県		
㈱アポロ電子本店	福岡市中央区城内6-21	092-712-0755
㈱波夢人E-TEK博多店	福岡市博多区吉塚3-31-61	092-623-6049
ワールド通信	福岡市博多区麦野5-8-38	092-593-1030
パル通信	福岡市西区拾六町3-11-20リバーサイドビル102	092-891-4370
Ｉ・Ｍ電子パーツ	田川市松原1区2-13	0947-44-6950
北九電子	北九州市八幡西区田町2-4-13	093-641-7178
㈱波夢人小倉店	北九州市小倉北区片野4-3-9	093-951-1916
ハムラヂオ	北九州市小倉北区霧ケ丘3-5-17	093-931-4949
㈲福岡ハムセンター	春日市伯玄町2-1	092-571-4949
㈲キャット通信	粕屋郡須恵町大字須恵462-78	092-933-4499
筑豊ハムセンター	嘉穂郡筑穂町元吉319-4	0948-72-0461
佐賀県		
ハムセンター佐賀店	佐賀市高木瀬東3-1-2	0952-33-9488
長崎県		
㈱ハムセンター長崎	長崎市岩川町13-20	0958-46-1950
サセボハムセンター	佐世保市福石町8-4	0956-31-8649
熊本県		
㈱熊電総業	熊本市江津1-745	096-379-9999
ハムショップ イズミ	鹿本郡鹿本町下高橋34-1	0968-46-4900
大分県		
㈲ウエーブ・シティ	大分市高松1-7-3	0975-51-6022
㈱サンアイ無線	大分市東浜1-13-9	0975-58-3232
㈲ハムショップ大分	大分市大字永興字三ケ田191-1	0975-44-8186
㈲佐伯テクニカ	佐伯市鶴岡町2-1-23	0972-23-4949
㈲佐伯無線	佐伯市鶴岡古市区２班	0972-23-7777
宮崎県		
ハムショップ宮崎	宮崎市神宮1-28-2	0985-22-8224
宮崎ハムセンター	宮崎市東大淀1-3-45	0985-52-4949
㈲原口無線	都城市大王町1-7	0986-25-2169
㈲ハラ無線	延岡市出北5-15-1	0982-33-4949
永田無線	小林市大字細野117-1	0984-23-3492
㈲岩本無線	日向市大字日知屋11202-18	0982-52-3917
鹿児島県		
㈱馬場電機	鹿児島市鷹師1-6-19	099-252-2676
㈱明昭堂Ｍ１	鹿児島市西千石町12-28	099-225-2020
クマデンin川内	川内市大小路町18-4	0996-22-4949
㈲南九州電子パーツ	川内市大小路町29-13	0996-22-4281
㈲大寿	鹿屋市寿5-27-4	0994-40-1222
シバタムセン	名瀬市井根町21-11	0997-53-8488
㈲ＣＱステーション	姶良郡隼人町東郷1321-64	0995-42-7211
キヨハラ無線	大島郡瀬戸内町古仁屋春日	09977-2-0592
沖縄県		
八重山無線	石垣市登野城214	09808-2-9967
㈱沖縄電子	宜野湾市字大山3-3-9	098-898-2358

JAIA
日本アマチュア無線機器工業会
〒170-0002
東京都豊島区巣鴨1-10-5 第2川端ビル2階 TEL.03-3944-8611

中古パソコン総合情報サイト

U-COMNET



欲しい情報 一発検索！！

http://www.u-comnet.com

U-COMNETは貴方のPCライフを応援します。

愛読者のみなさんに感謝の気持ちを込めて。

ビッグサマープレゼント2001！

三洋製　動画クリップデジタルカメラ（DSC-MZ1)を、5名様にプレゼント！

応募方法：　http://www.u-comnet.comのなかのプレゼント申し込みページから
アンケートに必要事項を入力して、送信（無料）ボタンを押してください。
抽選で5名様にデジタルカメラをプレゼント
平成13年9月10日締め切りです。
※当選の発表は、商品の発送をもってかえさせて頂きます。



U-COMNET 中古パソコン価格検索サイト

- ●当サイトは、中古パソコン、周辺機器、PCパーツ等の情報を一発検索できるサイトです。
- ●ページ内、中古パソコンショップ一覧では、全国の中古パソコン取り扱いショップ情報を閲覧できます。
- ●ページ内、外神田7丁目広場では、目的別BBS、チャットルームを用意して、読者のフォーラムとして解放しています。ぜひご利用ください。

※下記の広告募集中とは、ページ内の情報提供者様を対象にしています。

http://www.u-comnet.com/
広告募集中
U-COMNET
中古パソコン価格検索サイト

お問い合わせは

ユーコムネット事務局　〒114-0024　東京都北区西ヶ原1-65-2　　E-mail　webmaster@u-comnet.com



体験実習開催日程

7月26日(木)
27日(金)
28日(土)
8月 2日(木)
3日(金)
4日(土)
8月23日(木)
24日(金)
25日(土)
●午後1時より



お申込みは電話・FAX・Eメール・はがきで!!年令・希望学科を明記。

●参加は申込み順となりますので、希望日を早めに決めてください。もし定員に達した時は、次の開催日となることがあります。
●体験実習では、自分の適性や学院の雰囲気を知ることができます。また、個別に進路相談や入学相談なども受付けています。

人気のある学校は、ここがちがう!

❶初心者でも、2年後の卒業時には、ゲームデザイナーとして制作会社へ就職できます。
❷創立以来、低廉適正な学費を設定。年間費用84万円(すべての学費)14年度入学生。
❸生徒数の増加で毎年最新マシンを購入設置し、実習しているのが評価されています。
❹先生は業界に精通している、現役ゲームデザイナーが指導し、担任制をとっています。
❺クラスの中で4～5名のチームを組みゲームを制作。教室は7時まで開放している。

■就職抜群!学生寮完備。朝日・毎日・読売新聞等奨学金制度(学費全額免除・給与月額12万円支給)三井住友銀行教育ローン有。学院見学随時可⇨日時をTELにて要予約

2002年4月生募集中

ゲームデザイナー学部
全日制2年・定員250名

ゲームデザイン科
プログラム科
グラフィック科
キャラクターデザイン科
デジタルサウンド科



文部省認可CGARTS協会認定CG教育校/社団法人・日本漫画家協会賛助会員/社団法人・日本映画テレビ技術協会会員

日本ゲームソフト学院

〒101-0061 東京都千代田区三崎町3−4−9 ☎03-3261-8211

入学案内書 はがき(住所・氏名・年令・電話番号明記)か、電話で上記まで(無料)

FAX 0120−194−750 http://www.tag.co.jp/ E-mail:ngs@tag.co.jp

中古通販

PC NET ピーシーネット

在庫保有数日本一
(常時10,000台)

TEL.03-3733-1441

通販営業時間／月～金 9:30～19:00 土・日・祝日 11:00～19:30

www.prins.co.jp/

楽天市場店 http://www.rakuten.co.jp/pcnet/ ネットプライスモール http://www.netpricemall.com/
ジャストネット http://email.justnet.ne.jp/shop/pcnet/ ビッグローブショッピングワールド http://shopping.biglobe.ne.jp/

最新情報は電話かネットでチェック!! 急げ!!

この夏最大のチャンス
かならず **ネットをチェック!!**

夏休み大セールキャンペーン 実施中

何が特価になるか ヒ・ミ・ツ!
ミステリーサービス!!
電話で 毎日要チェック! ネットで

インターネットでは毎回お得な企画がいっぱい!!

3日間保証セール だから安い!! 期間限定タイムセール

さらにお求めやすくなりました!

幕張店 年中無休で営業中

[安心の全品保証付]　1万円以上…1ヵ月保証　1万円以下…1週間保証

※本商品は添付品（OSを含む）は付属しておりません。
※クリーニングご希望の方、別途¥1,000が必要です。
※中古パソコン販売時の在庫定義が変わりました。

今後在庫とは、従来品に加え、動作上支障のないキズ・天板やけ・カバー等の欠品がある物や、キーボード等が補充（他メーカー含）されている商品も、「在庫」としてお取り扱いさせていただきます。

DOS/V

注文No.	商品	価格
U1462	Celebris FX5200M(DEC)(MP(200)/32MB/2.1GHD/CD8)・・・・・・・・・・・・・・・・・・	¥17,800
U1812	Celebris FX-2 5233M(DEC)(MP(233)/32MB/2.1GHD/CD12/外装焼け・キズ/PCIX2/ISAX2/USBX2)	¥14,800
U1331	VENTURIS FX5200S(DEC)(Pen(200)/48MB/2.5GHD/CD4/外装焼け)・・・・・・	¥12,800
U8342	Aptiva(2190-22L)(IBM)(K6-2(450)/64MB/6.4GHD/CD40/外装キズ/マウス無し/省スペース型/USBx2)	¥31,800
U8344	Aptiva(2190-27N)(IBM)(K6-2(500)/60MB/10GHD/CD/スタンド無し/省スペース型/PCIX3/USBX2)	¥34,000
U2799	PC300GL(6272-C4J)(液晶セット)(IBM)(MP(166)/32MB/2.5GHD/CD/13.6DSTN(1024x768)/LAN/外装焼け/USBx2)	¥29,800
U8401	PC350(6587-JP6)(IBM)(Pen(200)/64MB/1.6GHD/CD6/外装焼け/擦り傷/32MB増設/ISAX2/共有X3/USBX2)	¥15,800
U2532	PC300GL(6272-A4J)(IBM)(Pen(166)/32MB/2.5GHD/CD/外装焼け/共有X2/USBX2)	¥8,000
C0096	PC-VS20CS/5CA1(NEC)(MP(200)/32MB/3.2GHD/CD20/FAX56K/付属品付)	¥17,800
U8518	PC-MA26D/F(液晶一体型)(NEC)(Pen II (266)/64MB/4.3GHD/CD24/15TFT(1600X1200)/LAN/外装焼け/一体型/USBX2)	¥67,800
U8519	PC-MA43H/FL(液晶一体型)(NEC)(Celeron(433)/64MB/6.4GHD/CD24/15TFT(1280X1024)/LAN/外装焼け/一体型/USBX2)	¥79,800
U8548	PC-MA45L/R(NEC)(K6-2(450)/64MB/6.4GHD/CD40/LAN/外装塗装剥げ/マイクロタワー型/PCIX3/USBX2)	¥30,800
U1898	DESKPRO 2000/5200/2500(COMPAQ)(Pen(200)/32MB/2.5HD/CD増設済み/PCIX2/ISAX2/共有X1)	¥14,800
U2992	DESKPRO 2000/6200(COMPAQ)(PPro(200)/32MB/2.5GHD/CD8/汚れ)・・・・	¥16,000
SH486	DESKPRO 4000S/5233X(COMPAQ)(MP(233)/32MB/2.1GHD/CD20)・・・・・・	¥17,000
V0169	DPENSF6400C/6.4/CDSW(COMPAQ)(Celeron(400)/64MB/6.4GHD/CD24/省スペース型)	¥30,800
U8537	OPTIPLEX GXA EM(DELL)(Pen II (233)/32MB/3.2GHD/CD/高さ16.5cm)・・・	¥16,800
U8382	OPTIPLEX GX1P(DELL)(Pen III (500)/128MB/10GHD/CD40/LAN/タワー型/PCIX3/ISAX2/共有X2/USBX2)	¥40,800
U8551	GATE WAY PROFILE(液晶一体型)(GATEWAY)(K6-2(400)/64MB/4.3GHD/DVD/15.1TFT/外装焼け/PCカバー無し/一体型/USBX2)	¥69,800
U8556	E5200JP(GATEWAY)(Pen II (400)/256MB/9GHD/CD/ファイルカバー無し/CDデバイスドライバー無し/ミニタワー型47cm/USBX2)	¥32,800
U2934	Vectra VE5/200-4(HP)(MP(200)/32MB/2.1GHD/CD24/外装焼け/ゴム足無し/擦り傷)	¥16,800
U1687	FMV-9D30A61(富士通)(MP(233)/32MB/3.2GHD/CD20/外装焼け/PCIX2/ISAX2)	¥19,800
U1045	FMV-S5237(富士通)(Pen II (233)/32MB/3.2GHD/CD24)・・・・・・・・・・・・・・・・・	¥21,800
U1736	FMV-6233D8(FMV-623D8C5)(富士通)(Pen II (233)/32MB/3.2GHD/CD16/外装焼け/PCIX2/ISAX2/USBX3)	¥21,800
U1364	FMVT4267M1.3(富士通)(Pen II (266)/32MB/4.3GHD/CD17/マイク・スピーカー無し/外装焼け/PCIX3/ISAX1/共有X1/USBX2)	¥25,800
U1375	MEBIUS MN-835-H23(液晶一体型)(シャープ)(Pen II (233)/32MB/3.2GHD/CD20/15TFT(1024X768)/外装焼け(全体的)/キーボード非純正/USBX2)	¥64,800
U8559	VAIO PCV-L520/BP(液晶セット)(ソニー)(Pen III (500)/128MB/13GHD/CD24/DVD/15TFT(1024X768))	¥99,800

DOS/Vノート

注文No.	商品	価格
U3570	HiNote CT475(DEC)(486DX4(75)/4MB/340HD/9.5TFT)・・・・・・・・・・・・・・・・・・・	¥9,800
U6732	ThinkPad300D(2635-3AJ)(IBM)(MP(150)/32MB/2.1GHD/CD8/SVGA/12.1TFT)	¥38,000
U3198	ThinkPad365X(2625-3J9)(IBM)(Pen(120)/24MB/810HD/SVGA/擦り傷/外装焼け/10.4TFT)	¥16,800
U6747	ThinkPad390X(2626-G0J/G5J/GNJ)(IBM)(Celeron(400)/64MB/6.4GHD/CD無し/SVGA/14.1TFT/USBX1)	¥82,800
U6726	ThinkPad560E(2640-40J)(IBM)(MP(166)/48MB/4GHD/SVGA/キーボード文字剥げ/12.1TFT)	¥37,800
U6559	ThinkPad600(2645-41J)(IBM)(Pen II (233)/64MB/3.2GHD/CD24/XGA/外装擦り傷/外装塗装剥げ/13.3TFT/USBX1)	¥64,800
U6351	PC-VP16CSAA1(NEC)(MP(166)/32MB/2.1GHD/CD20/SVGA/外装キズ/塗装剥げ/12.1TFT)	¥39,800
U3679	PC-AL20CTSAAF1(NEC)(MP(200)/32MB/2.1GHD/CD20/SVGA/外装塗装剥げ/キーボード文字剥げ(10ケ以下)/12.1TFT)	¥52,800
U3768	PC-VA23C/WX(NEC)(MP(233)/64MB/3GHD/CD20/XGA/13.3TFT)・・・・・・・・	¥63,800
U6731	PC-LT23D/54C.D(NEC)(Pen II (233)/64MB/4.3GHD/CD24/XGA/外装塗装剥げ/14.1TFT/USBX1)	¥62,800
U6650	PC-VA23D/WS(NEC)(Pen II (233)/64MB/2.1GHD/CD20/SVGA/外装塗装剥げ/擦り傷/12.1TFT/USBX1)	¥59,800
U6661	PC-VA23DTX(NEC)(Pen II (233)/64MB/2GHD/CD20/XGA/擦り傷/14.1TFT/USBX1)	¥59,800
U6655	PC-VA23DTX(NEC)(Pen II (233)/64MB/2GHD/CD20/XGA/外装塗装剥げ/汚れ/擦り傷/外装ヒビ/14.1TFT)	¥51,800
U6549	PC-VA26H/WSAA(NEC)(Celeron(266)/64MB/4.3GHD/CD/SVGA/12.1TFT)	¥57,800
U3606	PC-VA26CBSAYU34(NEC)(MP(266)/64MB/3GHD/CD無し/モデム/SVGA/外装塗装剥げ/10.4TFT)	¥57,800
U6672	PC-VA30H/WS(NEC)(Celeron(300)/64MB/4.3GHD/CD24/SVGA/外装塗装剥げ/外装キズ/12.1TFT/USBX1)	¥62,800
U6673	PC-VA33D/WX(NEC)(Pen II (333)/32MB/4.3GHD/CD24/XGA/14.1TFT/USBX1)	¥72,800
U6734	PC-LW333D74C,D6(NEC)(Pen II (333)/64MB/6.4GHD/CD24/XGA/14.1TFT/USBX1)	¥75,800
U6724	PC-VA40H/WS(NEC)(Celeron(400)/64MB/4.3GHD/CD24/SVGA/12.1TFT/USBX2)	¥77,800
U6741	PC-VA45J/AX(NEC)(Pen II (450)/256MB/12GHD/CD24/XGA/14.1TFT/USBX2)	¥94,800
U6274	FMV-5120NU2/W(FMV-52NU2W5.1共通)(富士通)(Pen(120)/32MB/1GHD/SVGA/外装擦り傷/11.3TFT)	¥15,800
U3799	FMV-50NUWH(FMV-5100NU/W)(富士通)(Pen(100)/32MB/810HD/SVGA/外装焼け/11.3TFT)	¥12,000
U3941	FMV-53NU5S(富士通)(Pen(133)/32MB/1GHD/CD4/SVGA/外装キズ/11.2TFT)	¥29,000
U3575	FMV-53NA6W(FMV-5133NA6/W)(富士通)(MP(133)/32MB/1.6GHD/CD20/SVGA/外装焼け・キズ/12.1TFT)	¥34,800
U3640	FMV-56NA5X(富士通)(MP(166)/32MB/2GHD/CD10/XGA/外装キズ/13.3TFT)	¥46,800
U6485	FMV-5ANP8W5(FMV-5200NP8/W)(富士通)(MP(200)/32MB/2.1GHD/CD20/SVGA/LAN/拡張ステーション標準添付/12.1TFT)	¥52,800
U6752	FMV2NS6X0,5,6(FMV-6233NS2/X)(富士通)(Pen II (233)/64MB/3.2GHD/CD/XGA/擦り傷/汚れ/ネジカバー無し/13.3TFT/拡張ステーション付/USBX1)	¥59,800
U6755	DynaBook SATELLITE310CDT(PA1262U)(東芝)(MP(200)/64MB/2.1GHD/CD16/SVGA/英語版キーボード/12.1TFT/USBX1)	¥47,800
U6744	DynaBook PA1271C9(東芝)(MP(200)/64MB/4GHD/CD20/SVGA/PCカバー欠品/開閉爪折れ/12.1TFT)	¥51,800
U6087	PAP301JC(東芝)(MP(266)/64MB/4.3GHD/SVGA/モデム/PCカバー無し/擦り傷/10.4TFT)	¥63,800
U6575	DynaBook SS6000(PAP600JA)(東芝)(MP(266)/64MB/4.3GHD/SVGA/CD(拡張ステーション)/外装塗装剥げ/擦り傷/12.1TFT/USBX1)	¥69,800
U6577	PAP302JA.B(東芝)(MP(300)/64MB/4.3GHD/SVGA/擦り傷/PCカバー無し/USポートカバー無し/10.4TFT/USBX1)	¥72,800
U6688	DynaBook PAP333J8,W(東芝)(Celeron(333)/64MB/4.3GHD/XGA/外装塗装剥げ/PCカバー無し/11.3TFT/USBX1)	¥70,800
U6681	DynaBook PAS403JB,C,D(東芝)(Celeron(300)/64MB/4.3GHD/CD24/XGA/擦り傷/14.1TFT/USBX1)	¥72,800
U6680	DynaBook PAS254N5,S,K(東芝)(K6-2(333)/64MB/4.3GHD/CD24/SVGA/13DSTN/USBX1)	¥62,800
U6520	DynaBook PAS403J.C.B.D(東芝)(Celeron(333)/64MB/4.3GHD/CD24/XGA/14.1TFT)	¥74,800
U6729	DynaBook PAS206N8.N5.NW.NN.Nk(東芝)(K6-2(366)/64MB/4.3GHD/CD24/SVGA擦り傷/12.1TFT)	¥64,800
U6711	PAS255N5,NS,NK,N8,NW,NM(東芝)(Celeron(366)/64MB/6.4GHD/CD24/汚れ/14.1TFT)	¥79,800
U6678	DynaBook SS7020X2(PP702E363A56)(東芝)(Pen II (366)/64MB/6.4GHD/XGA/PCカバー無し/外装塗装剥げ/13.3TFT/USBX1)	¥84,800
U6646	DynaBook Satellite2710 P50/4CA(東芝)(Pen III (500)/64MB/12GHD/CD24/XGA/モデム/擦り傷/外装塗装剥げ/14.1TFT/USBX1)	¥128,000
U6258	AL-N3T516J5(パナソニック)(MP(166)/32MB/1.6GHD/CD14/SVGA/擦り傷/外装塗装剥げ/12.1TFT)	¥44,800
U3145	MEBIUS PC-A455(シャープ)(Pen(133)/16MB/1GHD/CD/開閉ツメ折れ/キーボード文字剥げ(2ケ)/12.1TFT)	¥24,800
U6695	GATE WAY 2000 SOLO5150SE(GATEWAY)(Pen II (233)/64MB/6GHD/CD/14.1TFT)	¥67,800

バッテリーパックは消耗品につき保証対象外とさせていただきます。

98デスクトップ

注文No.	商品	価格
U2760	PC-9821CR13/TA(Pen(133)/16MB/850HD/CD4/14.4Kモデム/TV/TFTカラー液晶/擦り傷/一体型)	¥34,800
U1440	PC-9821XA20/W30(Pen(200)/64MB/3GHD/CD8/PCM/2FDD仕様/外装キズ・焼け)	¥24,800
U8350	PC-9821XA20/W30(Pen(200)/96MB/3GHD/CD8/PCM/64MB増設/外装焼け・キズ/PCIX2)	¥23,800
U1636	PC-9821XC16/S5A3,S5B3(Pen(166)/32MB/1.6GHD/CD8/外装キズ/外装塗装剥げ/外装焼け)	¥12,800
U1998	PC-9821XC200/S7A3,B3(MP(200)/64MB/2GHD/CD8/LAN/外装キズ・焼け/32MB増設済み/PCIX1/USBX2)	¥25,800
U8354	PC-9821RA266/W・D30R(NEC)(Pen II (266)/64MB/3GHD/CD16/32MB増設/PCIX2/ISAX3)	¥38,800

98ノート

注文No.	商品	価格
U3572	PC-9821LS150/S1C2(MP(150)/32MB/1.GHD/CD10/SVGA/AC非純正/12.1TFT)	¥46,800
U3787	PC-9821LS150/S14(MP(150)/32MB/1.4GHD/CD10/VGA/擦り傷/PCカバー無し/12.1TFT)	¥45,800
U6748	PC-9821NA7/H3(40MB/2GHD)(Pen(75)/40MB/2GHD/PCM/10.4TFT)・・・・	¥14,800
U6538	PC-9821NA9/H8(Pen(90)/24MB/810HD/CD4/PCM/外装焼け・キズ/セカンドバッテリー付き(消耗品につき保証対象外)/10.4TFT)	¥17,800
U3386	PC-9821NA12/H10(Pen(120)/16MB/1GHD/PCM/SVGA/CD4/10.4TFT)・・・	¥27,800
U3275	PC-9821NR12/D10(Pen(120)/16MB/1GHD/外装擦り傷/12.1DSTN)・・・・・・	¥22,800
U6490	PC-9821NR13/S14(Pen(133)/32MB/1.4GHD/SVGA/擦り傷/12.1TFT)・・・・	¥27,800
U3852	PC-9821NR150/X14F(MP(150)/32MB/1.4GHD/CD10/モデム/XGA/リアカバー無し/12.1TFT)	¥49,800
U6738	PC-9821NR166/X30F(MP(166)/64MB/3GHD/CD10/XGA/モデム/擦り傷/13.3TFT)	¥59,800
U1647	PC-9821NW133/D14C.D共通(MP(133)/32MB/1.4GHD/CD11/SVGA/12.1DSTN)	¥24,800

バッテリーパックは消耗品につき保証対象外とさせていただきます。

MAC

注文No.	商品	価格
U9346	Performa 6310(本体のみ)(PPC603e(100)/16MB/1.2GHD/CD4/外装焼け)・・・	¥8,000
U9351	PowerMac 7300/166(PPC604e(160)/16MB/2GHD/CD12/外装キズ/外装焼け(全体))	¥12,800
U9344	PowerMac 7300/180(PPC604e(180)/16MB/2GHD/CD12)・・・・・・・・・・・・・	¥21,000
U0287	PowerMac 7500/100(PPC601(100)/16MB/1GHD/CD/外装焼け/ファイルカバー無し)	¥7,800
U9190	PowerMac 7500/100(PPC601(100)/32MB/1GHD/CD/外装焼け)・・・・・・・・	¥12,800
U9443	PowerMac 8100/100(PPC601(100)/32MB/700HD/CD2/外装焼け/外装キズ)	¥19,800
U2054	PowerMac 8500/120(PPC604(120)/32MB/2GHD/CD)・・・・・・・・・・・・・・・	¥23,000
U9175	PowerMac 8500/132(PPC604(132)/64MB/1.2GHD/CD)・・・・・・・・・・・・・・	¥26,000
U9431	PowerMac 8500/150(PPC604(150)/49MB/1.2GHD/CD8)・・・・・・・・・・・・・	¥27,800
U9469	PowerMac 8500/180(PPC604e(180)/98MB/2GHD/CD8/外装焼け)・・・・・・	¥19,800
U9296	PowerMac 9500/132(PPC604(132)/32MB/2GHD/CD4)・・・・・・・・・・・・・・	¥36,800
U9301	PowerMac 9500/150(PPC604(150)/32MB/2GHD/CD4/外装焼け)・・・・・・・	¥41,800
U9427	PowerMac 9600/233(PPC604e(233)/80MB/4GHD/CD12/外装焼け)・・・・・	¥51,800
U9477	PowerMac G3/MT266(PPCG3(266)/64MB/6GHD/CD24/ZIP/外装焼け/汚れ)	¥68,800
U9476	PowerMac G4/400(PPCG4(400)/64MB/10GHD/DVD)・・・・・・・・・・・・・・	¥138,000

キーボードは別売となります。

PowerBook

注文No.	商品	価格
U3357	PowerBook 520C(68LC040(50)/12MB/320HD/コプロ無/9.5D-パッシブカラー)	¥16,800
U6133	PowerBook 1400CS/117(PPC603e(117)/16MB/770HD/CD6/SVGA/11.3DSTNカラー)	¥39,800

ディスプレイ

注文No.	商品	価格
U4802	F14T1(NEC)(14.1TFT:1024X768max/3列15pin)・・・・・・・・・・・・・・・・・・・	¥34,800
U4580	F15T1(NEC)(15TFT:1024X768max/外装焼け/外装キズ/3列15pin)・・・・・・	¥39,800
U4718	DV15A2(NEC)(15マルチ:1024X768max/3列15pin)・・・・・・・・・・・・・・・・・・・	¥7,500
U4505	DV17C4(NEC)(17マルチ:1280X1024max/外装焼け/3列15pin)・・・・・・・・・	¥12,800
U4819	Ultra SCANP780(DELL)(17マルチ:1600X1200max/外装キズ/3列15pin)・	¥12,000
SH711	CPD-20SF3(ソニー)(20マルチ:1280X1024max/外装キズ・焼け/3列15pin/5BNC)・	¥25,800
U4553	D2825(HP)(15マルチ:1024X768max/3列15pin)・・・・・・・・・・・・・・・・・・・・・	¥6,800
SH268	CPD-17SF8R(ソニー)(17マルチ:1280X1024max/0.25mm/外装焼け・キズ/3列15pin)	¥11,800
SH446	CPD-17GS(トリニトロン)(ソニー)(17マルチ:1280X1024max/0.25mm/外装キズ・焼け/3列15pin)	¥11,800
U4799	RDT-141R(三菱)(14TFT:1024X768max/約26万色/3列15pin)・・・・・・・・・・	¥38,800
U4469	RD-17V(ダイヤモンド)(三菱)(17マルチ:1280X1024max/0.25mm/外装キズ・焼け/3列15pin/直付)	¥11,800
SH708	MF-8517B(フラット)(飯山電機)(17マルチ:1280X1024max/0.28mm/外装キズ・焼け/2列15pin/5BNC)	¥12,000
U4146	MT-8617E(飯山電機)(17マルチ:1600X1200max/0.26mm/外装焼け・キズ/3列15pin/5BNC)	¥12,800
U4742	RST13X(AKIA)(13.3TFT液晶:1024X768max/3列15pin)・・・・・・・・・・・・・・	¥29,800
U4773	PC-DT3130(日立)(13.3TFT:1024X768/外装焼け/3列15pin)・・・・・・・・・・・・	¥28,000
U4827	PC-DT3150(日立)(15.5スーパークリアカラー:1024X768/外装焼け/3列15pin)	¥45,800
U4824	VX700(GATEWAY)(17マルチ:1600X1200max/外装キズ/3列15pin)・・・・・	¥14,800
U4720	CV-173(TOTOKU)(17マルチ:1280X1024max/外装焼け/3列15pin)・・・・・・	¥13,000

プリンタ

注文No.	商品	価格
U5209	PR-D700LX(NEC)(ドットインパクトプリンター:A3)・・・・・・・・・・・・・・・・・・・・・	¥59,800
U5220	PR-L2800N(NEC)(レーザー:LT～A3)・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	¥35,000
U5152	LBP-2030(キヤノン)(レーザー)・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	¥48,000
U5219	XJ-GNP(富士ゼロックス)(A4対応～ハガキ)・・・・・・・・・・・・・・・・・・・・・・・・・	¥6,800

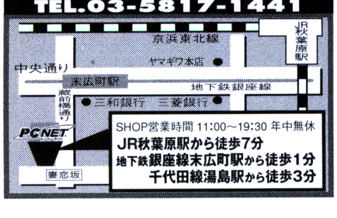
トナー等は消耗品につき標準添付されておりません。別途、ご確認下さい。

商品の状態により価格の変動がある事がございます。
詳しくはTELかネット上でご確認ください。

万一、品切れの際はご容赦ください。（7月25日現在）

こちらの在庫リストは通販専用となります。店舗在庫は各店舗へご来店の上ご確認ください。

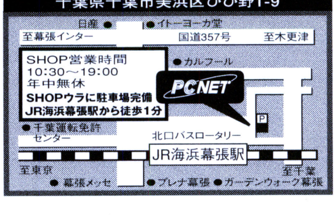
秋葉原店
千代田区外神田6-5-12 偕楽ビル新末広2F（1Fジョナサン）
TEL.03-5817-1441



大阪日本橋店
浪速区難波中2-1-7 なんば東ビル
TEL.06-4396-1401



幕張店
千葉県千葉市美浜区ひび野1-9



パシフィック ネット

株式会社パシフィックネット
本社 〒144-0051 東京都大田区西蒲田7-10-1
相互西蒲田ビル TEL.03-3733-1441 FAX.03-3733-1440
通販営業時間／月～金 9:30～19:00 土・日・祝日 11:00～19:30

計測機器を各ショップのみで販売中。
詳しくは各ショップまでお問い合わせください。

iモードでショップ最新情報検索
http://www.prins.co.jp/i/

◆ 定期購読●直接購読のしおり ◆

弊社刊行の雑誌は，ご自宅直送方式の年間定期購読をお勧め申し上げます。スピーディーで確実，そしてお得な定期購読をぜひご利用下さいませ。

また　ご希望の雑誌・書籍が，ご近所の書店にない場合は右の振替用紙にて直接ご注文下さい。

■ 年間定期購読料（消費税込料金となっております。）

電　気　店	￥16,000	マイコンBASICマガジン	￥6,120
電波新聞縮刷版（CD-ROM版）	￥96, 000		

■ 年間購読の特典

①特別号も通常の定価でご購読いただけます。
②電気店は13冊分の購読ができます。

◆確実な振替口座を利用してご送金下さい◆

送金手数料が不要で，確実な郵便振替でご送金をお願い申しあげます。手続きは簡単です。右の郵便振替用紙に下記の必要事項を記入し，郵便局窓口に必要金額と共にご提出いただくだけですみます。

①購読申込書に必要事項を記入ください。なお１年間，或は２年間購読かをはっきりご記入ください。
②払込通知票，払込票に送金合計金額と住所，氏名をご記入ください。払込通知票と払込票は絶対に切りはなさないようにしてください。なお，必ずフリガナをご記入下さい。
③弊社への御意見・御希望がございましたら，通信欄へご記入下さい。
④すべて記入が終了しましたら，案内文のみ切り離して郵便局窓口に金額を添えて提出してください。

∵振替によるご送金は，着本までに約２週間ほどかかりますのでお含みおきください。お急ぎの場合は，現金書留に，右購読申込書を同封してご送金ください。

この受領証は、郵便局で機械処理をした場合は郵便振替の払込みの証拠となるものですから大切に保存してください。

ご注意

この払込書は、機械で処理しますので、口座番号及び金額を記入する際は、枠内に丁寧に記入してください。

また、下部の欄（表面及び裏面）を汚したり、本票を折り曲げたりしないでください。

（郵政事業庁）

払込取扱票

02 東京

通常払込料金 加入者負担

口座番号（右詰めにご記入ください）

0 0 1 5 0 - 3 - 5 1 9 6 1

金額　千 百 十 万 千 百 十 円　※

加入者名　株式会社 電波新聞社

料金　特殊取扱

通信欄

※定期購読誌名		購読期間	送金額（送料当社負担）
	*新規 継続	年　月号より　年間	円
御希望誌・ソフト・グッズ名		冊数	送金額（送料共）
			円
			円

フリガナ 送付先住所 □□□-□□□□	産業分類	所属又は職種
フリガナ 送付先御氏名	電話 TEL	年令　才 / 性別　男・女

払込人住所氏名（郵便番号　　）※（電話番号　－　－　）

受付局日附印

各票の※印欄は、払込人において記載してください。

裏面の注意事項をお読みください。郵政事業庁（私製承認 東 第1005号）

切り取らないで郵便局にお出しください。

記載事項を訂正した場合は、その箇所に訂正印を押してください。

払込票兼受領証

口座番号　0 0 1 5 0 - 3　通常払込料金加入者負担

右詰めにご記入ください

5 1 9 6 1

加入者名　株式会社 電波新聞社

金額　千 百 十 万 千 百 十 円　※

払込人住所氏名　※

料金（消費税込み）　円

受付局日附印

特殊取扱

切り取って郵便局へお出し下さい。

電波新聞社出版物一覧

（消費税込料金となっております）

2001.7 現

最新エレクトロニクス用語事典	5,300円	〒380円	DOS／V制作スペシャルNo.2～11	1,490円	〒34
半導体・ICのすべて	2,520円	〒310円	F-BASIC V6.3プログラミングのすべて	2,835円	〒31
ビデオ技術のすべて	2,100円	〒310円	ネットワーク対応データベース構築ガイド	2,854円	〒31
TVのすべて改訂版	1,988円	〒310円	パソコン認定試験合格対策試験問題集3・4級	2,100円	〒31
CD技術のすべて	2,039円	〒310円	デジタル1種〈技術科目〉最新テキスト	2,205円	〒34
MDのすべて	1,400円	〒310円	デジタル3種受験マニュアル	3,000円	〒34
センサーのすべて	2,000円	〒310円	アナログ・デジタル全資格〈法規科目〉	1,995円	〒34
改訂版DVD&DVC入門基本18章	1,529円	〒310円	シスアド受験スーパー記憶術	2,900円	〒34
増補改訂版デジタル放送がわかる本	1,600円	〒310円	基本情報受験スーパー記憶術	2,900円	〒34
増補改訂版・新ビデオ技術ハンドブック	2,500円	〒310円	特殊無線技士受験マニュアル・改訂新版	2,905円	〒34
新・まんが電気学入門	1,500円	〒310円	デジタルビデオカメラを使いこなす本シリーズ ビクター・ソニー・パナソニック・シャープ・キャノン	各 1,490円	各 〒31
新電気学入門講座	2,050円	〒310円			
新オシロ・ロジアナ入門講座	2,243円	〒310円	オールアバウト カプコン対戦格闘ゲーム1987～2000	2,100円	〒45
電子計測器ガイドブック	7,500円	〒450円	オールアバウト SNK対戦格闘ゲーム 1991～2000	2,310円	〒45
IT社会のビジネスプラットホーム	2,625円	〒310円	IT革命にどう生き残るか	2,205円	〒31

※上記以外にも新聞・出版物がございますので最寄りの本社、支局へお問い合わせ下さい。

東京本社 03-3445-8201（ダイヤルイン）
大阪本社 06-6203-3361
西部本社 092-431-7411
札幌支局 011-641-5591
仙台支局 022-227-7211
新潟支局 025-245-2526
関東総局 027-326-3206
松本支局 0263-36-2266
名古屋支局 052-261-4541
金沢支局 076-263-8661
京都支局 075-221-8021
広島支局 082-228-5581
高松支局 087-861-3111
鹿児島支局 099-226-3630

Q10　携帯・PHSを持っていますか？
①両方持っている ②携帯 ③PHS ④購入予定 ⑤予定なし

Q11　Q10で④の方のみ，どちらを購入する予定ですか？
①携帯 ②PHS ③両方 ④未定

Q12　使用中のプログラミング言語は？
※枠内に一つご記入ください

Q13　これから勉強したいプログラミング言語は？
※枠内に一つご記入ください

Q14　今月号の本誌で面白かった記事を三つ，下の表からお答えください

Q15　今月号の本誌でつまらなかった記事を三つ，下の表からお答えください

ご協力ありがとうございました。

■今月号の記事一覧

① エレクトロニック研究所訪問記
② 編集部の学校訪問記
③ プレステ2でプログラム！実践編
④ パソコンレクチャー ザ・ビギニング
⑤ Technical News Spot
⑥ はじめようMicrosoft Excelでの BASICプログラミング
⑦ IE＋DirectAnimation マルチメディア・プログラミング
⑧ 図解！Visual Basicゲーム作成術
⑨ 特選投稿プログラム・コーナー
⑩ CMM試聴室
⑪ 明日のスタープログラマー ＆Programmer's Room
⑫ Dr.Dのベスト・プログラム研究室
⑬ BASIC STUDIO パワフルゲーム工房入門
⑭ チャレンジ！ワンダーウィッチ
⑮ キミにもできる！「iアプリ」制作
⑯ 魁!!Dr.JのJAVA-WALK
⑰ Catlikeのもっと身近に プログラミング！
⑱ ロボ道場　パソコンでロボットをプログラミングしよう
⑲ レゴ道場　チャレンジ！ レゴ・マインドストーム
⑳ 松原拓也の自作一代記
㉑ テクノ・トイズ・ラボラトリ
㉒ よいこのLinux修行
㉓ iモード・ホームページ講座
㉔ 神奈川のりこのホームページとかに夢中
㉕ DTM Art Studio パソコンで音楽を楽しもう！
㉖ あくせすROOM
㉗ GIGA連射つつうらら
㉘ SOFTWARE MAKER HOT INFORMATION
㉙ 表紙
㉚ OVER FLOW

マイコンBASIC Magazine　プログラム投稿応募カード

言語名（音源名）					
プログラム名（曲名）					
氏　名		ペンネーム		年齢	歳
住　所	〒				
TEL	※連絡可能な時間帯	応募年月日	年 月 日		

キ リ ト リ 線

郵　便　は　が　き

141-8715

50円切手をお貼り下さい

（受取人）

東京都品川区東五反田1−11−15

電波新聞社

月刊 マイコンBASICマガジン 編集部

9月号 行

(フリガナ)		年齢	性別
ご氏名		歳	男・女
□□□-□□□□		電話	
ご住所	都・道 府・県		

職業：小学生／中学生／高校生／高専生／短大生／大学生／専門学校生／予備校生／アルバイト／公務員／会社員／その他（　　　　）

本誌を 毎月・時々・初めて 読む　何人で読みますか（　　）人

自分のホームページを　持っている・持つ予定・持っていない・予定なし

本誌のほかによく読む雑誌は？［　　　　］

ベーマガお楽しみプレゼント⑨ 応募欄（84ページ欄外）	希望番号	品名
	番	

キ リ ト リ 線

月刊マイコンBASICマガジン⑨ 愛読者カード

★OFコーナー宛 ♥あくせすROOM宛 ■質問・感想（本誌への掲載／可・不可）

ペンネーム：

■アンケート回答記入欄 ■ ※黒の筆記用具でわかりやすく，はっきりとご記入ください

Q 1		Q 2		〔メーカー名 〕				
Q 3		Q 4		Q 5				
Q 6								
Q 7					〔商品名 〕			
Q 8								
Q 9								
Q10		Q11						
Q12〔 〕		Q13〔 〕						
Q14				Q15				

キ リ ト リ 線

■アンケートにご協力ください。

※左の愛読者カード・記入欄に，以下の質問に該当する番号をご記入ください。複数回答の質問で，回答欄をすべて埋める必要はありません。

Q1　メインで使っているパソコンは？
①デスクトップ ②ノート ③持っていない

Q2　パソコンの種類は？（Q1で③以外の方）
①自作 ②ショップブランド ③メーカー製
※枠内にメーカー名も記入してください

Q3　メインで使っているOSは？（Q1で③以外の方）
①Windows 95 ②Windows 98 ③Windows Me ④Windows 2000 ⑤Windows NT ⑥Macintosh ⑦UNIX ⑧Linux ⑨その他

Q4　パソコン歴は？
①1年未満 ②1～2年 ③3～5年 ④6～10年 ⑤11年以上 ⑥なし

Q5　インターネット歴は？
①1年未満 ②1～2年 ③3～5年 ④6～10年 ⑤11年以上 ⑥なし

Q6　パソコンの主な使用目的は？（複数回答可）
①プログラミング ②ゲーム ③インターネット ④メール ⑤ミュージック ⑥学習 ⑦CG ⑧その他

Q7　これから買いたいと思うコンピュータ機器は？（複数回答可）
①パソコン本体 ②プリンタ ③スキャナ ④デジタルカメラ ⑤その他
※わかれば，商品名も記入してください

Q8　持っているゲーム機は？（複数回答可）
①プレイステーション（PSone含） ②プレイステーション2 ③ドリームキャスト ④ニンテンドウ64 ⑤ゲームボーイ ⑥ゲームボーイカラー ⑦ゲームボーイアドバンス ⑧ワンダースワン ⑨ワンダースワンカラー ⑩その他

Q9　これから買いたいと思うゲーム機は？（複数回答可）
①プレイステーション（PSone含） ②プレイステーション2 ③ドリームキャスト ④ニンテンドウ64 ⑤ゲームキューブ ⑥Xbox ⑦ゲームボーイカラー ⑧ワンダースワンカラー ⑨ゲームボーイアドバンス

※裏面へつづく→

■メモ欄…プログラム担当者・編集部へのご意見・ご感想など

圧倒的密度! 英雄伝説Ⅲ・Ⅳ・Ⅴ

新作

ガガーブトリロジー決定版



交響曲ガガーブトリロジー 完全新作新アレンジ

収録時間70分の超大作 第1部「朱紅い雫」、第2部「海の檻歌」、第3部「白き魔女」全18曲

イースⅠ・Ⅱアレンジ決定版

※ジャケットのデザインは変更される場合があります。

オリコン初登場 25位 (1990年9月・第3週集計)



(2枚組)

パーフェクトコレクション
イースⅠ・Ⅱ～米光亮全曲集

イースアレンジの決定版！パーフェクトコレクション
イース&イースⅡ米光亮全曲アレンジ完全復刻版！



(2枚組)

パーフェクトコレクション
イースⅠ・Ⅱ～スーパーアレンジ全曲集

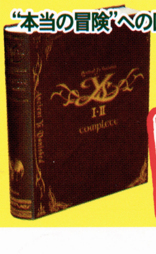
OVA「イース」エンディング曲「Endless History／南翔子」収録！
名盤パーフェクトコレクションイース&イースⅡ完全復刻版！

9月13日全国一斉発売！

[ファルコムミュージックCD取扱い店]【東京】メッセサンオー本店／ヤマギワソフト館／ラオックス ザ・コンピュータGAME館／石丸電気本店／石丸電気3号店／石丸電気ソフトワン／ゲーマーズ本店／ゲーマーズ秋葉原店／ゲーマーズスクエア店／ゲーマーズ水道橋店／ゲーマーズ池袋店／ゲーマーズ新宿店／HMV渋谷／タワーレコード渋谷店／ゲーマーズ立川店【大阪】ヤマギワソフト大阪店／ワルツ堂 日本橋第2店／ゲーマーズなんば店【北海道】そうご電器YES札幌【山形】シーガル十日町店【宮城】ヨドバシカメラ仙台店【埼玉】パルテック所沢店【神奈川】ゲーマーズ横浜店【長野】ラオックスヒナタコンピュータ館／ラオックスヒナタ松本コンピュータ館／ゲーマーズ長野店【愛知】コムロード上前津店／ゲーマーズ名古屋店【京都】ゲーマーズ京都店【兵庫】パソコンショップアリス【岡山】ベスト電器コンピュータウンくらしき【福岡】PC SHOP DENOH／ゲーマーズ博多店、他 ※取扱い店様、大募集！

■お求めは上記取扱い店またはカンタン便利なファルコム通販(falcom.comファルコムドットコム)もご利用下さい。

"本当の冒険"への回帰…。分かたれた物語は、今、ここに一つの神話となって甦る。



イースⅠ・Ⅱ完全版

PCゲーム定番

好評発売中！

WindowsMe/98/95(Windows2000対応)

詳しくは、
http://www.falcom.com/

Falcom®
NIHON FALCOM CORPORATION
〒190-0012 東京都立川市曙町1-14-13
☎042(527)6501 http://www.falcom.co.jp/

定価550円 本体524円 雑誌18361-09 T1118361090550

第二十巻第九号通巻二三二号 昭和五十七年十二月六日第三種郵便物認可 二〇〇一年八月二十五日印刷 二〇〇一年九月一日発行

電波新聞社 〒141-8715 東京都品川区東五反田一-十一-十五 電話〇三(三四四五)六一一一

2001

わかる！動かせる！
プログラムが組める雑誌

マイコンBASIC Magazine

9

プレステ2でシューティング・ゲーム制作！

http://www.basicmagazine.net/

電波新聞社